Fuzzy Set Methods for Object Recognition in Space Applications

Third Quarter Report

P. 81

James M. Keller

University of Missouri-Columbia

1/1/92 - 3/31/92

(NASA-CR-190388) FUZZY SET METHODS FOR OBJECT RECOGNITION IN SPACE APPLICATIONS Quarterly Report No. 3, 1 Jan. - 31 Mar. 1992 (Research Inst. for Computing and Information Systems) 81 p

N92-33093

Unclas

G3/61 0096749

Cooperative Agreement NCC 9-16
Research Activity No. SE.42

NASA Johnson Space Center
Information Systems Directorate
Information Technology Division

Research Institute for Computing and Information Systems
University of Houston-Clear Lake

INTERIM REPORT

RICIS Preface

This research was conducted under auspices of the Research Institute for Computing and Information Systems by James M. Keller of the University of Missouri-Columbia. Dr. Terry Feagin was the initial RICIS research coordinator for this activity. Dr. A. Glen Houston, Director of RICIS and Assistant Professor of Computer Science, later assumed the research coordinator assignment.

Funding was provided by the Information Technology Division, Information Systems Directorate, NASA/JSC through Cooperative Agreement NCC 9-16 between the NASA Johnson Space Center and the University of Houston-Clear Lake. The NASA technical monitor for this activity was Robert N. Lea, of the Software Technology Branch, Information Technology Division, Information Systems Directorate, NASA/JSC.

The views and conclusions contained in this report are those of the author and should not be interpreted as representative of the official policies, either express or implied, of UHCL, RICIS, NASA or the United States Government.

Third Quarter Report: January 1, 1992 to March 31, 1992

Fuzzy Set Methods For Object Recognition In Space Applications

Fixed-Price Subcontract NO. 088

Under Cooperative Agreement No. NCC 9-16

Project No. SE. 42

To:

UNIVERSITY OF HOUSTON-CLEAR LAKE
2700 Bay Area Boulevard
Houston, TX 77058

UHCL Technical Representative:

Dr. Terry Feagin

Principal Investigator:

James M. Keller
Electrical and Computer Engineering Department
University of Missouri-Columbia
Columbia, Mo. 65211
(314) 882-7339
ecekeler@umcvmb.bitnet

Introduction

For the third quarter of this research contract, we are going to report progress on the following four Tasks (as described in the contract):

2. Feature Calculation;

3. Membership Calculation;

4. Clustering Methods (including initial experiments on pose estimation);

5. Acquisition of images (including camera calibration information for digitization of model).

The report, as we have done in the past, consists of "stand alone" sections, describing the activities in each task. We would like to highlight the fact that during this quarter, we believe that we have made a major breakthrough in the area of fuzzy clustering. We have discovered a method to remove the probabilistic constraint that the sum of the memberships across all classes must add up to 1 (as in the fuzzy c-means). A paper, describing this approach is included (it is under review for the *IEEE Transactions on Fuzzy Systems*).

Feature Calculation

We have acquired images, digitized from video tape, and have begun the process of feature extraction and segmentation. We have concentrated on texture-based features and edge based features. On subsequent pages, we describe the 14 texture features which are calculate from the gray-tone spatial dependence matrices. We then show a typical image of the shuttle, with earth as background followed by images of various texture features extracted from the image. It is obvious from the resultant images that some of the features are good discriminators while others are quite poor. Following those images, we show the results of segmenting this image using the "threshold unit" approach described in the second quarter report.

The gray-tone spatial dependence matrix $P(i,j)$ is computed from a window with a size of $L_y \times L_x$. Denote d be the distance between the two pixels in the window, then we have,

$$P(i,j,d,0^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) | k-m=0,$$
$$|l-n| = d, I(k,l) = i. I(m,n) = j\}$$
$$P(i,j,d,45^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) | k-m=d,$$
$$l-n=-d \text{ or } (k-m=-d, l-n=d), I(k,l) = i. I(m,n) = j\}$$
$$P(i,j,d,90^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) || k-m| = d,$$
$$l-n=0, I(k,l) = i. I(m,n) = j\}$$
$$P(i,j,d,135^\circ) = \#\{((k,l),(m,n)) \in (L_y \times L_x) \times (L_y \times L_x) | k-m=d,$$
$$l-n=d \text{ or } (k-m=-d, l-n=-d), I(k,l) = i. I(m,n) = j\}$$

where # denotes the number of elements in the set.

The following notations are used to compute the 14 texture features.

Notation

$p(i,j)$ (i,j)th entry in a normalized gray-tone spatial-dependence matrix, $= P(i,j)/R$. R is a normalization factor.

$p_x(i)$ ith entry in the marginal-probability matrix obtained by summing the rows of $p(i,j)$. $= \sum_{j=1}^{N_g} P(i,j)$.

N_g Number of distinct gray levels in the quantized image.

$\sum_i$ and $\sum_j$ $\sum_{i=1}^{N_g}$ and $\sum_{j=1}^{N_g}$, respectively.

$$p_y(j) = \sum_{i=1}^{N_g} p(i,j).$$

$$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ i+j=k}}^{N_g} p(i,j), \qquad k = 2,3,\cdots,2N_g.$$

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ |i-j|=k}}^{N_g} p(i,j), \qquad k = 0, 1, \cdots, N_g - 1.$$

The 14 texture features are defined as following.

1) *Angular Second Moment:*

$$f_1 = \sum_i \sum_j \{p(i,j)\}^2$$

2) *Contrast:*

$$f_2 = \sum_{n=0}^{N_g-1} n^2 p_{x-y}(n)$$

3) *Correlation:*

$$f_3 = \frac{\sum_i \sum_j p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

where μ_x, μ_y, σ_x, and σ_y are the means and standard deviations of p_x and p_y.

4) *Sum of Squares: Variance*

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i,j).$$

5) *Inverse Difference Moment:*

$$f_5 = \sum_i \sum_j \frac{1}{1 + (i-j)^2} p(i,j).$$

6) *Sum Average:*

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i).$$

7) *Sum Variance:*

$$f_7 = \sum_{i=2}^{2N_g} (i - f_6)^2 p_{x+y}(i).$$

8) *Sum Entropy:*

$$f_8 = -\sum_{i=2}^{2N_g} p_{x+y} \log\{p_{x+y}(i)\}.$$

9) *Entropy:*

$$f_9 = \sum_i \sum_j \log(p(i,j)).$$

10) *Difference Variance:*

$$f_{10} = \text{variance of } p_{x-y}$$

11) *Difference Entropy:*

$$f_{11} = -\sum_{i=0}^{N_g-1} p_{x-y}(i) \log\{p_{x-y}(i)\}.$$

12), 13) *Information Measures of Correlation:*

$$\begin{aligned} f_{12} &= \frac{HXY - HXY1}{\max\{HX, HY\}}. \\ f_{13} &= (1 - \exp[-2.0(HXY2 - HXY)])^{1/2}. \\ HXY &= -\sum_i \sum_j p(i,j) \log(p(i,j)). \end{aligned}$$

where HX and HY are entropies of p_x and p_y. and

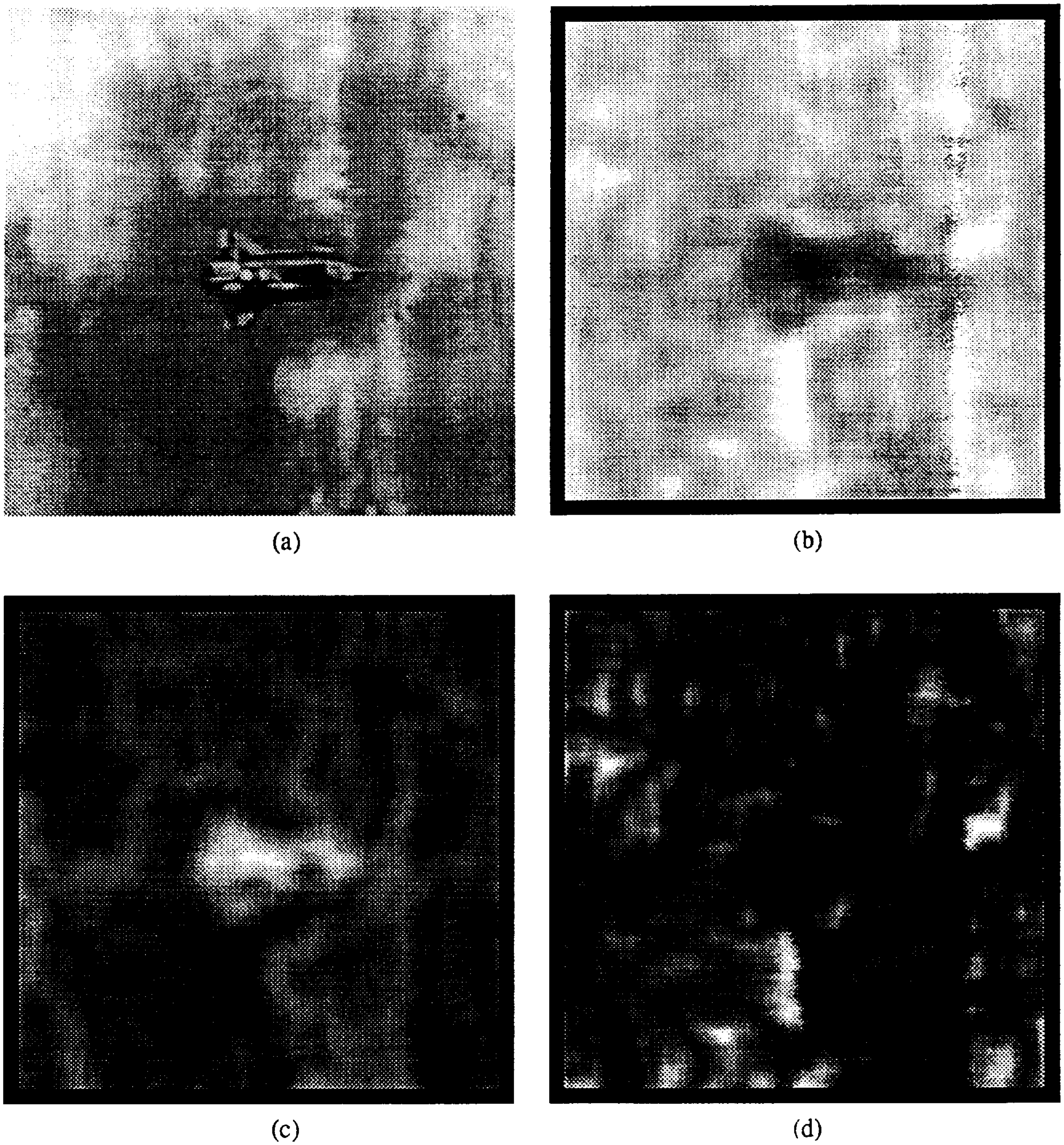
$$\begin{aligned} HXY1 &= -\sum_i \sum_j \log\{p_x(i)p_y(j)\} \\ HXY2 &= -\sum_i \sum_j p_x(i)p_y(j) \log\{p_x(i)p_y(j)\}. \end{aligned}$$

14) *Maximal Correlation Coefficient:*

$$f_{14} = (\text{Second largest eigenvalue of } Q)^{1/2}$$

where

$$Q(i,j) = \sum_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}.$$



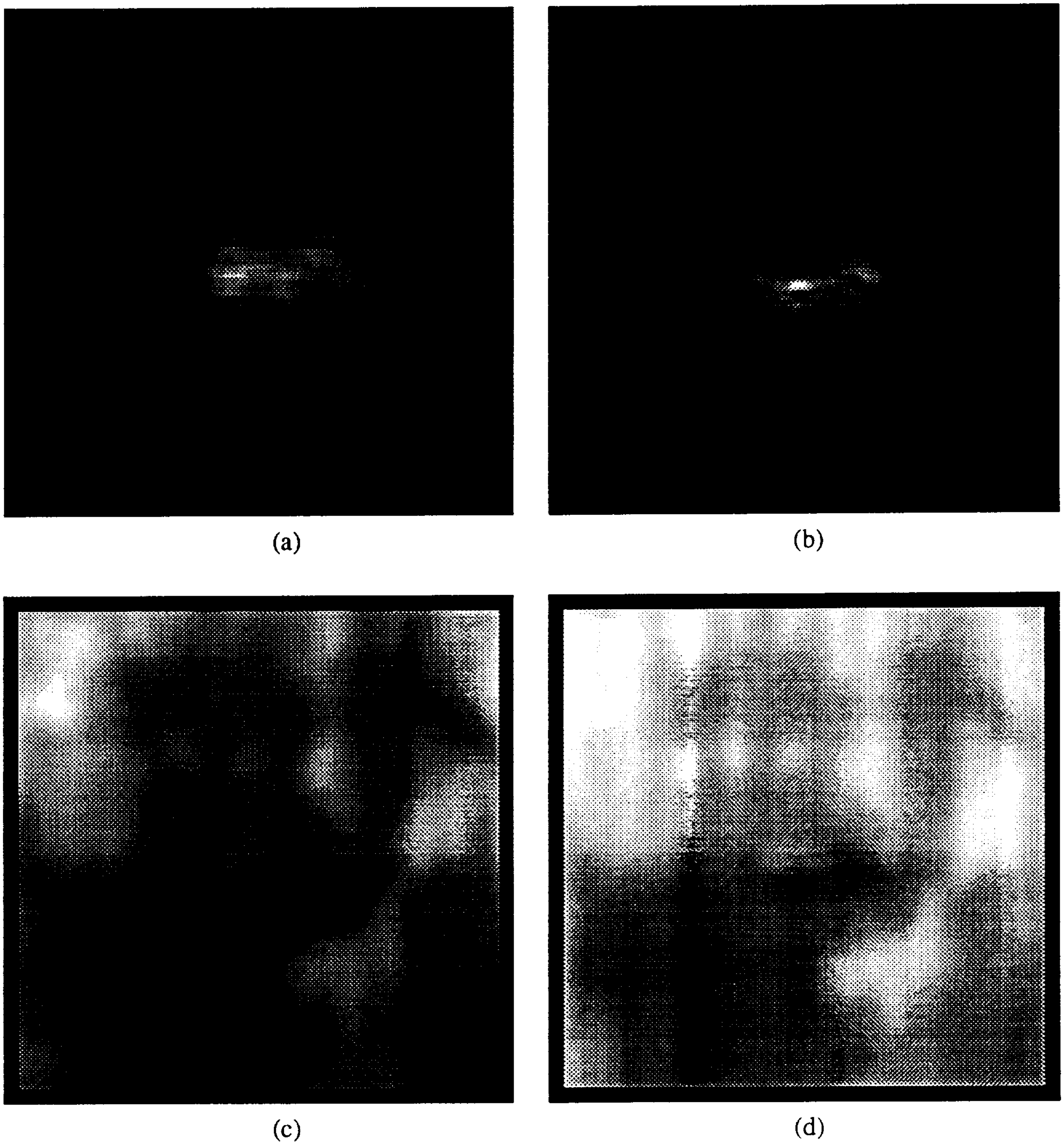
(a)

(b)

(c)

(d)

(a) original image
(b) homogeneity
(c) entropy
(d) angular second moment



(a)

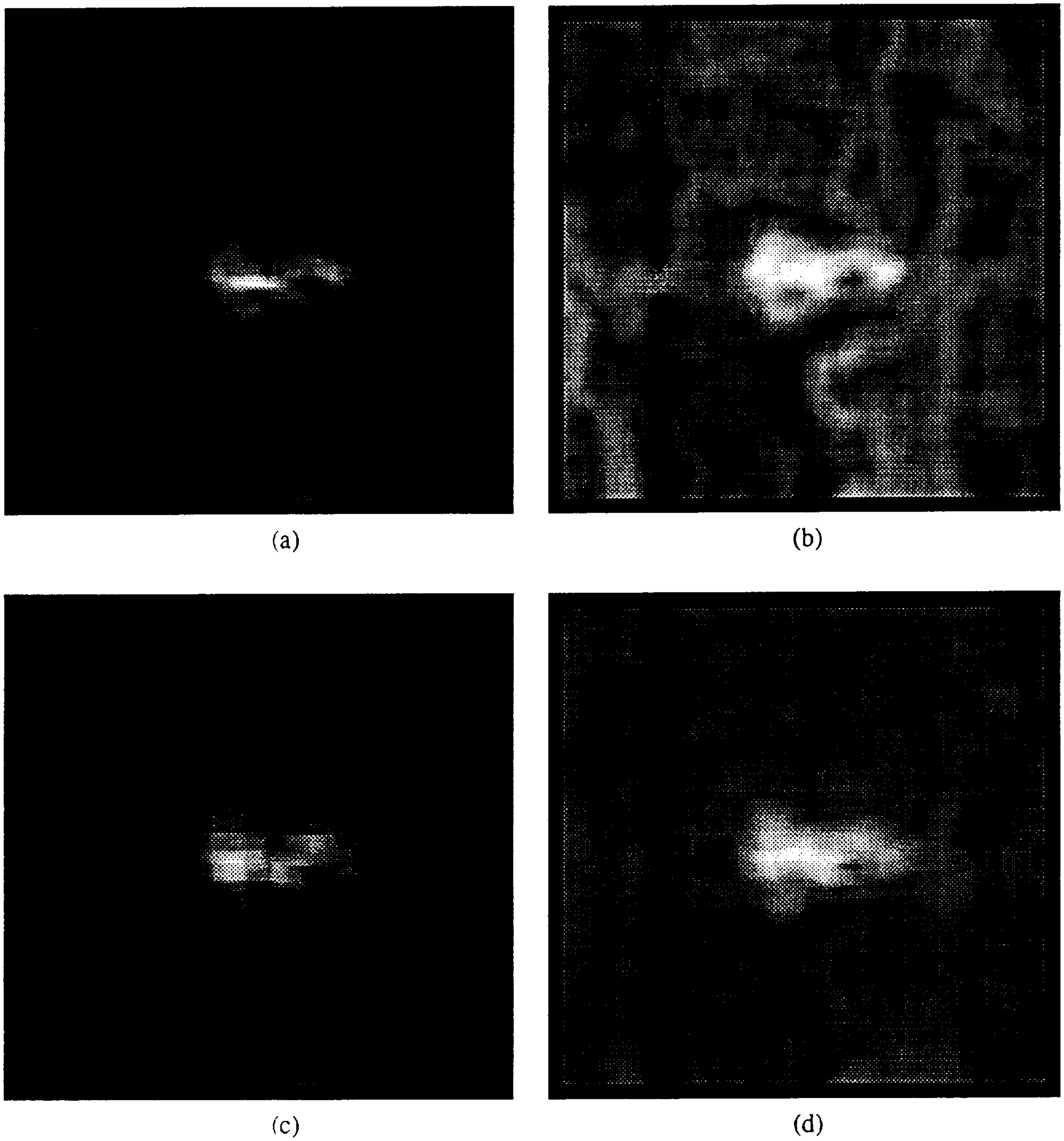
(b)

(c)

(d)

(a) contrast
(c) sum of squares

(b) correlation
(d) sum average



(a)

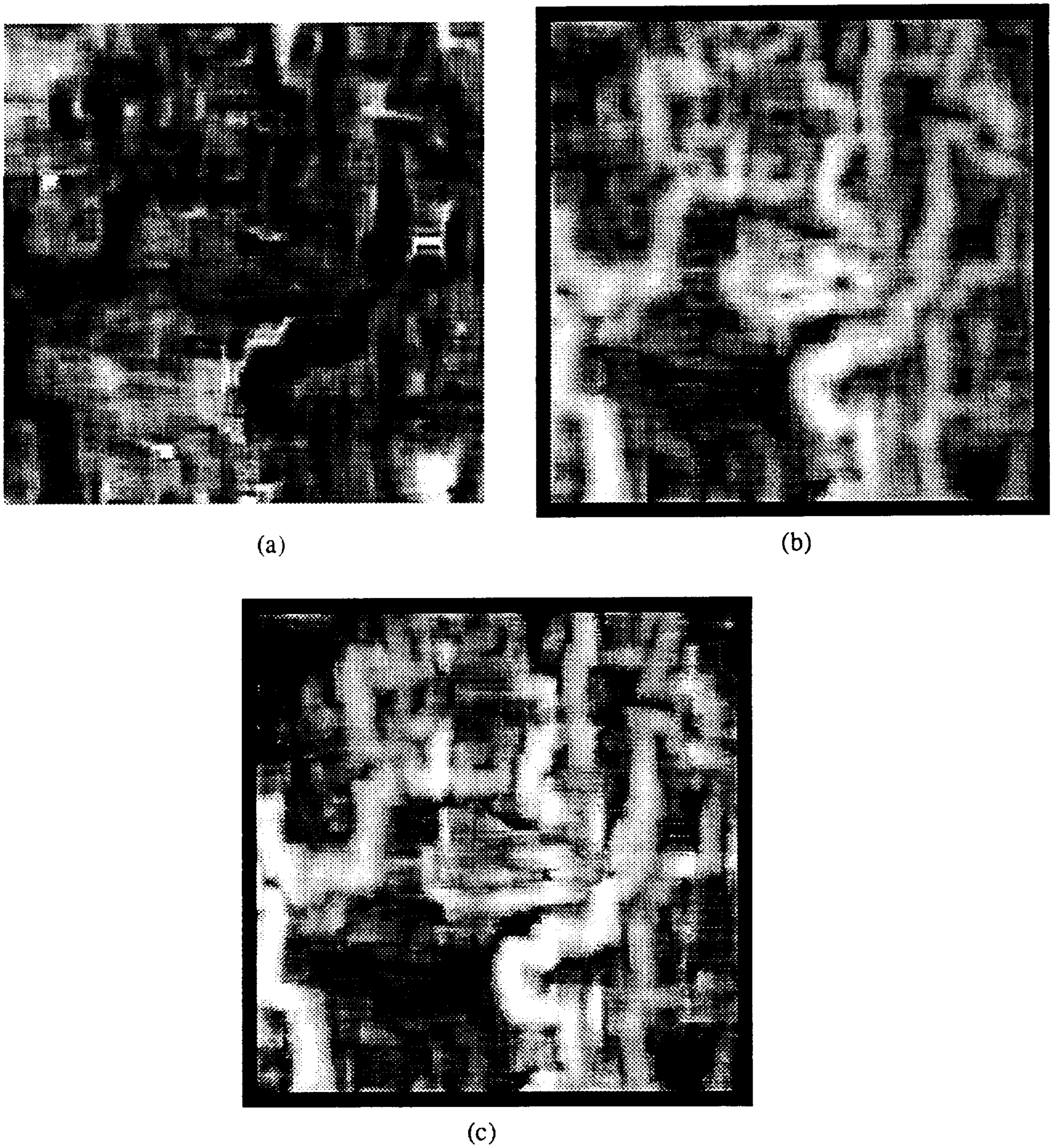
(b)

(c)

(d)

(a) sum variance
(c) difference variance

(b) sum entropy
(d) difference entropy



(a) (b)

(c)

(a) information measure of correlation, method 1
(b) information measure of correlation, method 2
(c) maximal correlation coefficient

Calculation of Membership Functions

Our work in this area has progressed nicely. We have designed and implemented numerous algorithms to generate membership values from a set of training data using histograms, results of fuzzy clustering, and heuristic definitions. We have also made progress in the transformation of "probability density functions" into possibility distributions for use in assigning membership values to individual points. The following report describes three methods of converting histograms of features into possibility distributions from which we can calculate membership function values for segmentation and recognition. The last example demonstrates these techniques on the "homogeneity" feature for the shuttle and background on our sample image.

Methods for Generating Membership Functions :

Fuzzy set theory has been used extensively in the literature for decision making, particularly in situations involving uncertain, vague and imprecise data supplied by heterogeneous sources. Many of these approaches involves the use of memberships (or degree of satisfaction of criteria). Thus membership functions play a crucial role in fuzzy set theoretic decision making.

There are three general approaches to constructing membership functions: i) heuristic methods ii) clustering methods, and iii) histogram-based methods. Heuristic methods assume that the shape of the membership functions is known(e.g. trapezoidal, triangular, and Gaussian). Heuristic methods have been used very successfully in control approaches. In computer vision, heuristic membership functions may be used to describe certain relational notions (such as above, below) and certain properties (such as lightness or darkness of a pixel value, position of a pixel, narrowness of a region). However, heuristic methods are not sufficiently flexible if one is to construct the membership functions from training data. This is because assumption on the shape of the membership function are too limiting. In such cases, fuzzy clustering methods and histogram based methods are more usefuL Clustering based methods use fuzzy clustering techniques to obtain a fuzzy partition of the training data. The membership values generated by the fuzzy partition are used for decision making. Several fuzzy clustering techniques exist in the literature, however, this approach will not be discussed here. We now describe membership generation techniques based on histograms of feature (training) data.

In image processing applications, histograms have been traditionally treated as probability distributions. In pattern recognition, many methods exist for estimating pdf's from samples. Since probabilities represent relative frequencies, it is reasonable to assume that if we have a huge number of samples that represent the ensemble whose area has been normalized to 1. Thus, methods that transform probabilities to possibilities (membership values) can be used to generate membership functions from histograms. To our knowledge, there are three ways to construct membership functions from given probability density functions. All of these methods assume that we have probability density functions at hand before these methods are applied, or we can approximate them as histograms.

Method 1:

This method suggested by D. Dubios and H. Prade[1] is based on probability/necessity measure theories.

Let $X= \{x_i \mid i= 1,....n\}$ to be a universe of discoures. The x_i's are ordered such that $p_1 \geq p_2 \geq ..... \geq p_n$, where $p_i = P(\{x_i\})$, $\sum_{i=1}^{n} p_i = 1$ and P is a probability measure. A_i denotes the set $\{x_1, x_2 ,......, x_i\}$. $A_0 = \varnothing$ by convention.

Definition 1. The degree of necessity of event A is the extra amount of probability of elementary events in A over the amount of probability assigned to the most frequent elementary event outside A. In order words

$$N(A) = \sum_{x_j \in A} max\,(pj - \max_{x_k \notin A} pk\ ,\ 0) \qquad (1)$$

Proposition 1. The set function N satifies the following axioms:

$$N(\varnothing) = 0\ ; \qquad N(X) = 1 \qquad (2)$$

$$\forall A,B \quad X \qquad N(A \cap B) = min(N(A),N(B)) \qquad (3)$$

Definition 2. Viewing $N(A)$ as the grade of impossibility of the opposite event $\bar{A}$ we can define the grade of possibility of A by

$$\forall A \quad X \qquad \Pi(A) = 1 - N(\bar{A}) \qquad (4)$$

where $N(\bar{A})$ is defined by (1)

Proposition 2. The set function Π defined by (4) verifies the axioms.

$$\Pi(\varnothing) = 0\ ; \qquad \Pi(X) = 1: \qquad (5)$$

$$\forall A,B \qquad \Pi(A \cup B) = max(\Pi(A),\ \Pi(B)) \qquad (6)$$

Hence Π is a possibility measure in the sense of Zadeh.

Denoting , $\pi_i = \Pi(\{x_i\})$, we have $\pi(A) = \max_{x_i \in A} \pi_i$

so that Π and N are completely specified through the possibility distribution $\{\pi_i \mid i = 1,\ldots,n\}$ which can be viewed as a normalized fuzzy set.

The π_i's are easily obtained from the pi's since

$$\pi_i = 1 - N(X - \{x_i\}) = 1 - N(A_{i-1})$$

$$= 1 - \sum_{j=1}^{i-1} (p_j - p_i) \qquad \text{for } i > 1 \tag{7}$$

and $\pi_1 = 1$ (normalization). Using $\sum_{i=1}^{n} p_i = 1$ we get

$$\forall i = 1,\ldots\ldots,n \quad \pi_i = ip_i + \sum_{j=i+1}^{n} p_j \tag{8}$$

The equation(8) gives us a way to generate membership functions from given probability density functions.

It is easy to see from (8) that for $i=1, \ldots ,n-1$

$$\pi_i - \pi_{i+1} = i(p_i - p_{i+1}) \tag{9}$$

and thus

$$\pi_i = \pi_{i+1} \Leftrightarrow p_i = p_{i+1}, \qquad \pi_i > \pi_{i+1} \Leftrightarrow p_i > p_{i+1}$$

i.e., the possibility distribution and the probability density have the same shape.

Proposition 3. If the probability assignment p maps on the possibility distribution π via (8) then

$$\forall A \quad N(A) \leq P(A) \leq \Pi(A)$$

Therefore, the possibility distribution they defined satisfies Zadeh's probability/possibility consistency principle.

Proposition 4. The possibility distribution π_i is greater or equal to normalized probability distbution. that is,

$$\forall\, i = 1,.....,n \qquad \pi_i \geq \frac{p_i}{p_1(=p_{max})}$$

proof : From Eq(8), $\pi_i = ip_i + \sum_{j=i+1}^{n} p_j$

consider that $\frac{p_1}{p_i}\, \pi_i = \frac{p_1}{p_i} ip_i + \frac{p_1}{p_i} \sum_{j=i+1}^{n} p_j = ip_i + \frac{p_1}{p_i} \sum_{j=i+1}^{n} p_j$

Note that $ip_1 \geq \sum_{j=1}^{i} p_j$ and $\frac{p_1}{p_i} \sum_{j=i+1}^{n} p_j \geq \sum_{j=i+1}^{n} p_j$

Therfore, $\frac{p_1}{p_i}\, \pi_i \geq 1$ so, $\pi_i \geq \frac{p_i}{p_1}$

Experimental Results :

We applied this methodl to some simple pdf's, for example, linearily decresing, triangle, trapezoid, and normal pdf's. As shown in figure 1-1, for the linear parts of pdf's we have the corresponding possibility distribution in the form of quadratic equation as expected. One might refer to Appendix A for the closed form solutions of possibility distributions corresponding to these simple pdf's . The shape of possibility distribution for a normal pdf is much similar to the original one.

We applied this to a homogeneity feature of a space shuttle image in Figure 2-1(b). In this experiment, we constructed histograms of a object(shuttle) and a background by sampling pixels from Figure 2-1(b), and smoothed histograms by a binomial window with window size 11. Figure 2-2 shows that the smoothed histogrmas of object and background, and their corresponding possibility distributions. Note that the possibility distributions are computed from normalized histograms(i.e., pdf's).

Method 2 :

This method suggested by Klir[2] is based on uncertainty measure in probability and possibility theories. He claims that under the transformation, values p_i must correspond to values r_i for all i=1,.....,n by some appropriate scale and , in addition, the amount of information should be preserved.

In other words, the total amount of uncertainty in the probability distribution must be equal to the total amount of uncertainty in the possibility distribution.

Let $p = (p1,p2,.......,p_n)$ and $r = (r1,r2,.....,r_n)$ denote, repectively, probability and possitility distributions (defined on a finite set X with n elements) that do not contain zero elements and are ordered in such a way that $p_i \geq p_{i+1}$ and $r_i \geq r_{i+1}$ $\forall i$=1,2,.......n-1

That is, $p_i \in (0,1]$, $r_i \in (0,1]$ and $\sum_{i=1}^{n} p_i = 1.$

probabilistic measure of uncertainty is the well known shannon entropy.

$$H(p) = - \sum_{i=1}^{n} p_i \log2 \, p_i \tag{1}$$

In possibility theory, two type of uncertainty coexist, nonspecificity and discord ; their measures are

$$N(r) = - \sum_{i=2}^{n} p_i r_i \log2 \frac{i}{i-1} \tag{2}$$

$$D(r) = - \sum_{i=1}^{n-1} (r_i - r_{i+1}) \log2 \, [1 - i \sum_{j=i+1}^{n} \frac{r_j}{j(j-1)}] \tag{3}$$

respectively.

Hence, the requirement that information be preserved by the transformation is expressed by constraining the scaling between p and r by the equation.

$$H(p) = N(r)+D(r) \tag{4}$$

Klir contends that log-interval scal transformation is the only one that exists for all distributions and is unique .

Its form is

$$r_i = (p_i/p_1)^{\alpha}$$

where a is a positive constant determined by solving Eq (4) for given H(p).

From extensive computer experimentation, Klir conjectures that $\alpha \in [0,1]$. If the conjecture is true, than $r_i \geq p_i$ for all $i=1,\dots,n$ is guaranteed(probability-possibility consistency principle).

Experimental Results:

We applied this method 2 to previously defined pdf's. The results are also shown in figure 1-1. Our experiment shows that there are some cases where we may have $\alpha>1$ for the trapezoid and the normal pdf's.

The same homogeneity feature as used in method 1 was considered to compute the possibility distributions and all results are summarized in Figure 2-2. One can easily notice that the membership values computed by both methods (method 1 and 2) are quite similar.

Method 3:

The last method which we are investigating now is suggested by Civalar and Trussel[3]. It is based on an optimization technique to find optimal membership functions from a given pdf. They claim that in order to define a reasonable membership function, there are certaim conditions which can be imposed on the membership function to make the set have properties consistent with the user's subjective judgement and the underlying pdf. From a heuristic viewpoint, the elements which are most likely should have high membership values, however, the possibility distribution should be as specific as possibile. These requirements are quantitatively describved below:

1. $E(m(x) \mid x$ is distributed according to the underlying pdf$) \geq c$
where the confidence level c should be close to unity.

2. $0 \leq \mu(x) \leq 1$

3. $\int \mu^2(x)dx$ should be minimized to obtain a specific membership function.

The optimal membership function defined by these condition can be derived using constrainted optimization techniques. They found that the optimal membership function is given by

$$\mu(x) = \begin{cases} \lambda p(x) & \text{if} \quad \lambda p(x)<1 \\ 1 & \text{if} \quad \lambda p(x)\geq 1 \end{cases} \qquad (1)$$

where p(x) is the pdf or its estimate derived form the histogram of the feature used for defining the fuzzy set, and the constant -- is to be solved from

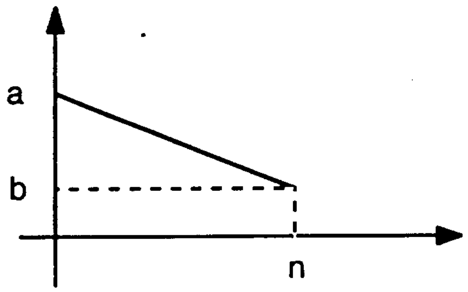
$$\lambda \int_{\lambda p(x)<1} p^2(\eta)d\eta + \int_{\lambda p(x)\geq 1} p^2(\eta)d\eta - c = 0 \qquad (2)$$

A interesting result they found is that the membership function corresponding to Gaussian with $c = 1/\sqrt{2}$ is a normalized Gaussian function with the highest value equal to 1.

APPENDIX A.

Some simple pdf's and its closed form solution of corresponding possibility distributions.

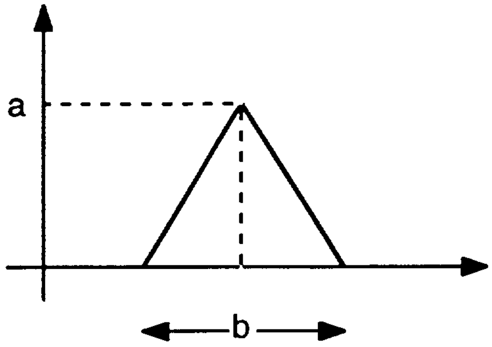
1. Linearly decreasing case:



$$p_i = [-\frac{a-b}{n-1} i + \frac{an-b}{n-1}]/c \text{ where } c = n(a+b)/2$$

$$\pi_i = \frac{a-b}{n(n-1)(a+b)}(-i^2 + i) + 1$$

2. Triangle case:

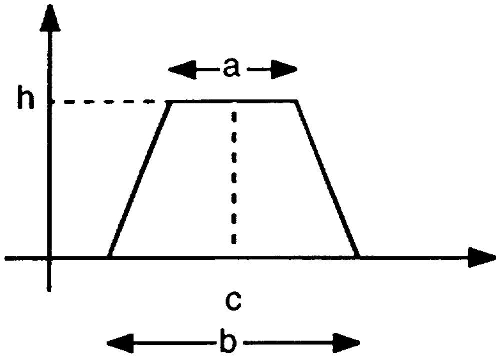


$$p_i = [-\frac{a}{b} i + \frac{a(a+1)}{b}]/c \quad \text{for i is odd number.}$$

$$p_i = [-\frac{a}{b} i + a]/c \quad \text{for i is even number where } c = \frac{a}{2}(n-1).$$

$$\pi_i = -\frac{i^2}{(n-1)^2} + \frac{2}{(n-1)^2} i + \frac{n(n-2)}{(n-1)^2} \quad \text{for i is odd number.}$$

3. Trapezoid case :



$p_i = [-\frac{h}{b-a} i + \frac{n(b+1)}{b-a}]/d$ for $i\,(\geq a+1)$ is odd number.

$p_i = [-\frac{h}{b-a} i + \frac{hb}{b-a}]/d$ for i(≥ a) is even number where $d = (a+b)h/2$.

$\pi_i = \frac{1}{(b+a)(b-a)}[-i^2 + 2i - n^2 + 2bn]$ for $i(\geq a+1)$ is odd number.

Appendix A. Probability distributions and corresponding possibilities generated by D. Dubios: Note that p_i's and π_i's are arranged in non-increasing order.

References.

1. George J. Klir, A principle of uncertainty and information invariance ,*Int. J. General Systems*, Vol. 17, pp.249-275 (1990).

2. Dider DUBOIS and Henri PRADE, Unfair conins and necessity measures : Towards a possibilistic interpretation of histograms, *Fuzzy sets and systems* 10, pp. 15-20 (1983).

3. M. Reha CIVANLAR and H. Joel TRSSELL, Constructing membership functions using statistical data, *Fuzzy sets and systems* 18 pp. 1-13 (1986).

4. L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, *Fuzzy sets and systems* 1 (1978).

5. J. DOMBI, Membershp function as an evaluation, *Fuzzy sets and systems* 35,pp.1-21 (1990).

6. M. DELGADO and S. MORAL, On the concept of possibility-probability consistency, *Fuzzy sets and systems* 21, pp.311-318 (1987).

7. R. Yager, Ed., Fuzzy set and possibility Theory, Recent Developents(Pergamon Press. Oxford, 1982) 90-97.

8. B. BHARATHI DEVI, Estimation of fuzzy memberships from histogrmas, *Information sciences* 35, pp.43-59 (1985).

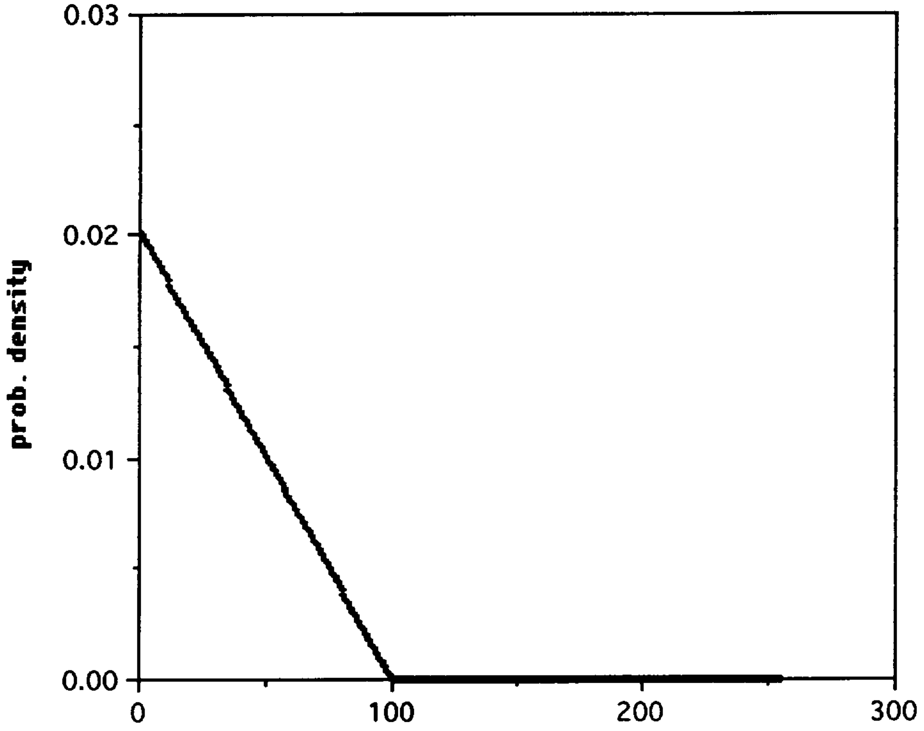


Figure 1-1: (a) linearily decreasing prob. density.

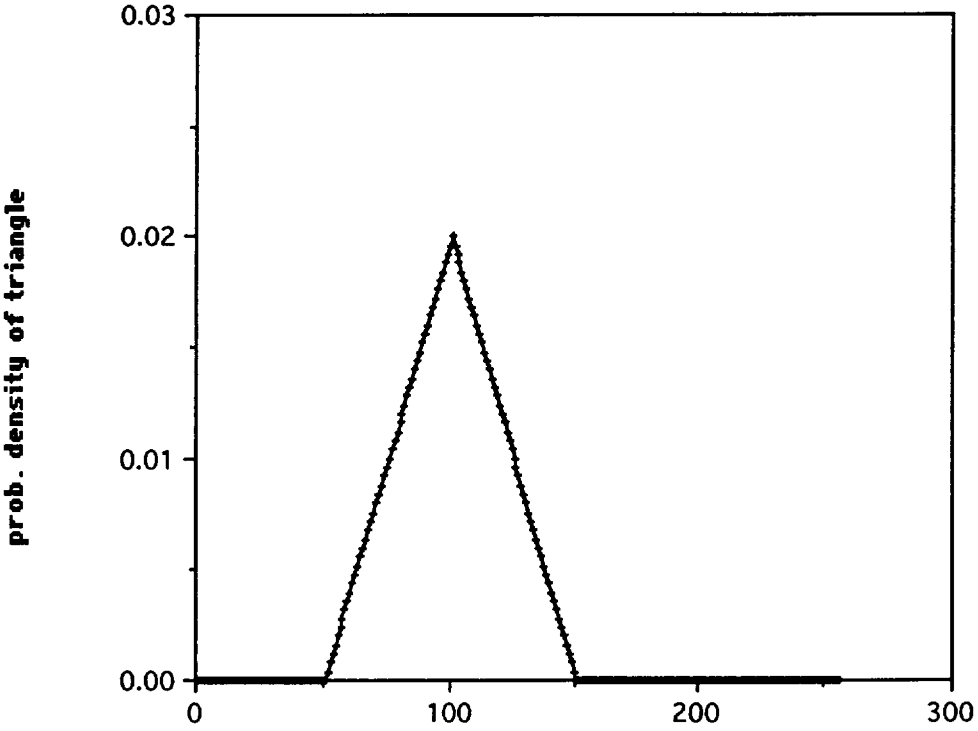


Figure 1-1 : (b) triangle pdf.

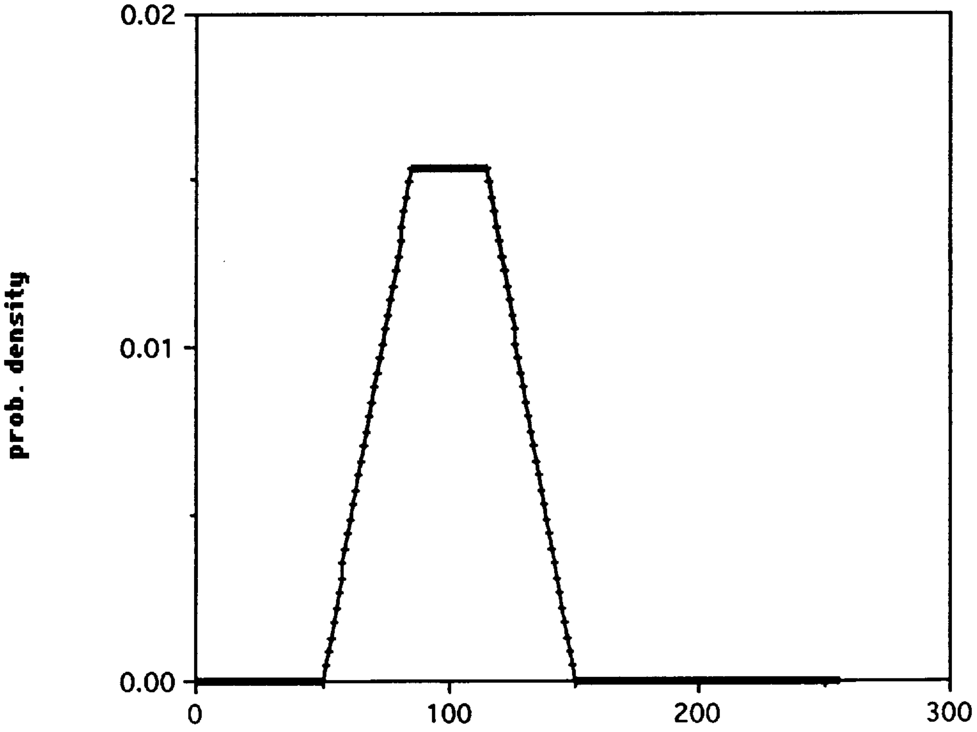


Figure 1-1 : (c) trapezoid pdf

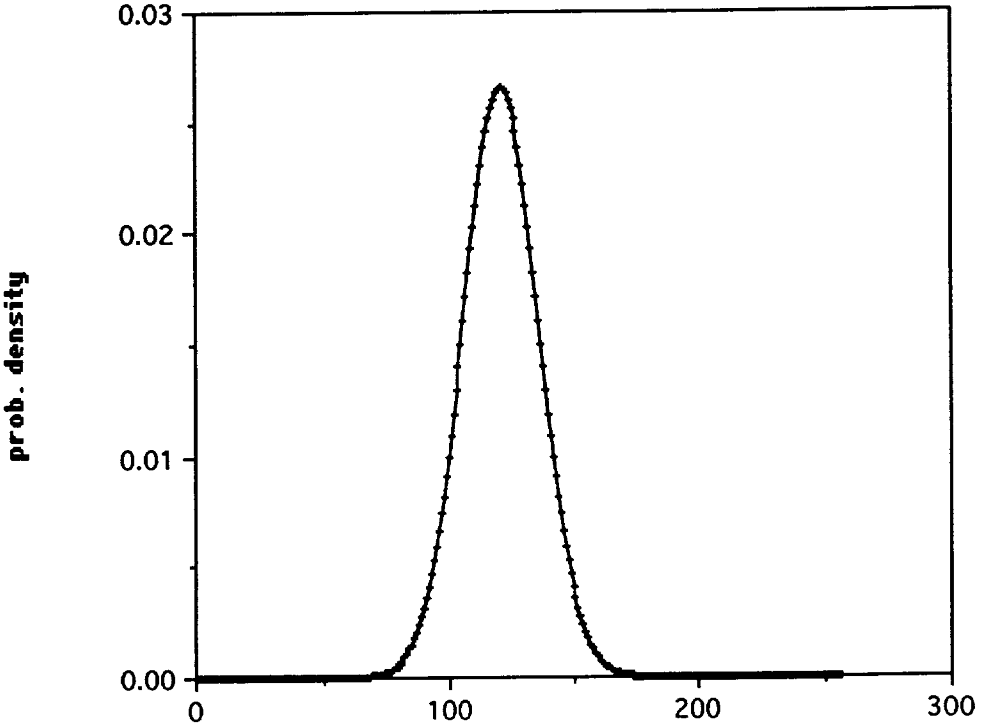


Figure 1-1 :(d) normal pdf.

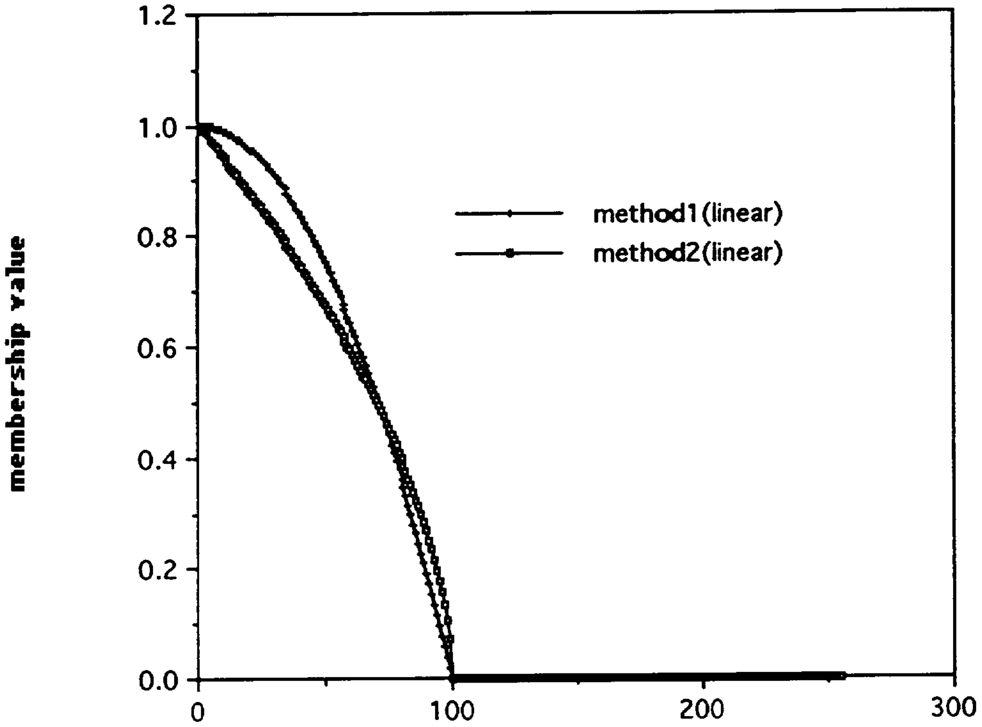


Figure 1-1 : (a') membership value of (a).

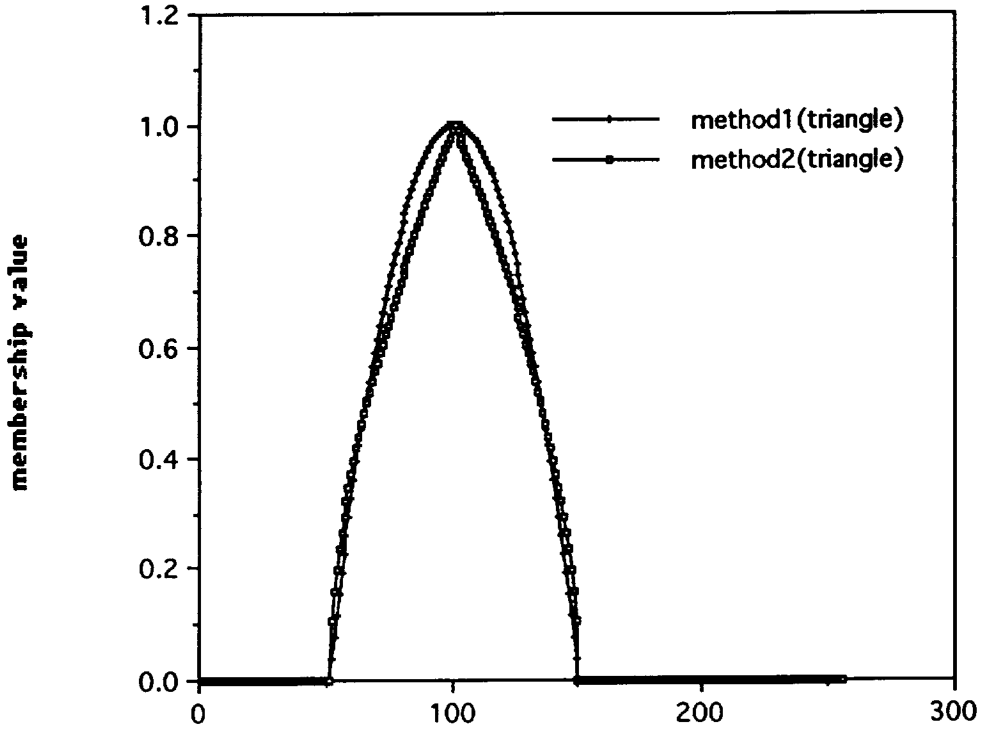


Figure 1-1 : (b') membership value of (b).

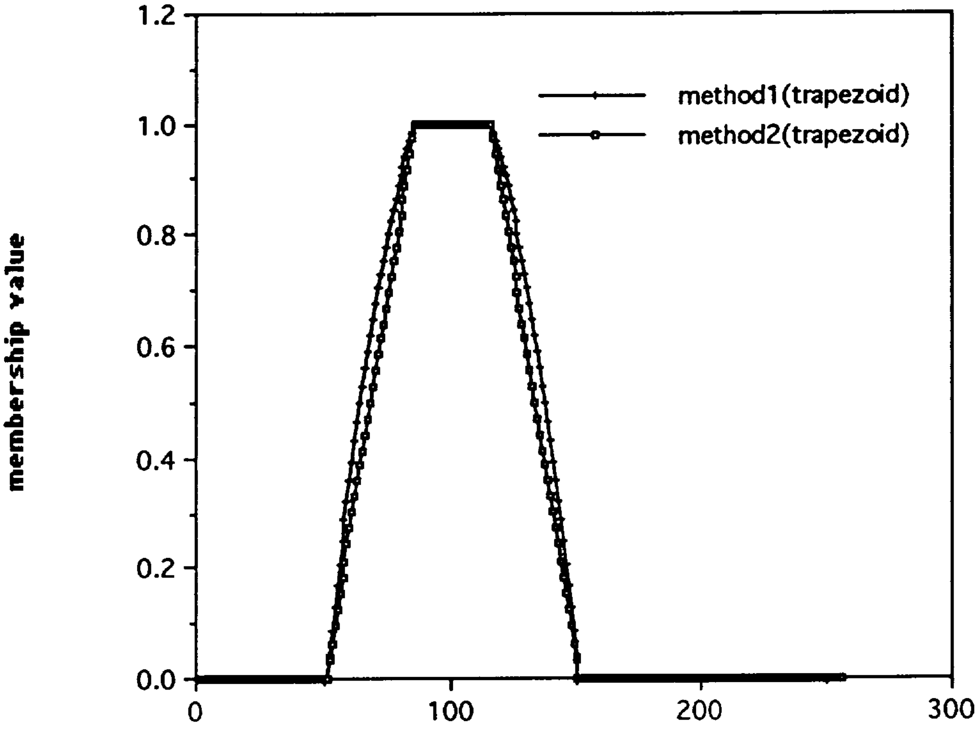


Figure 1-1 : (c') membership value of (c).

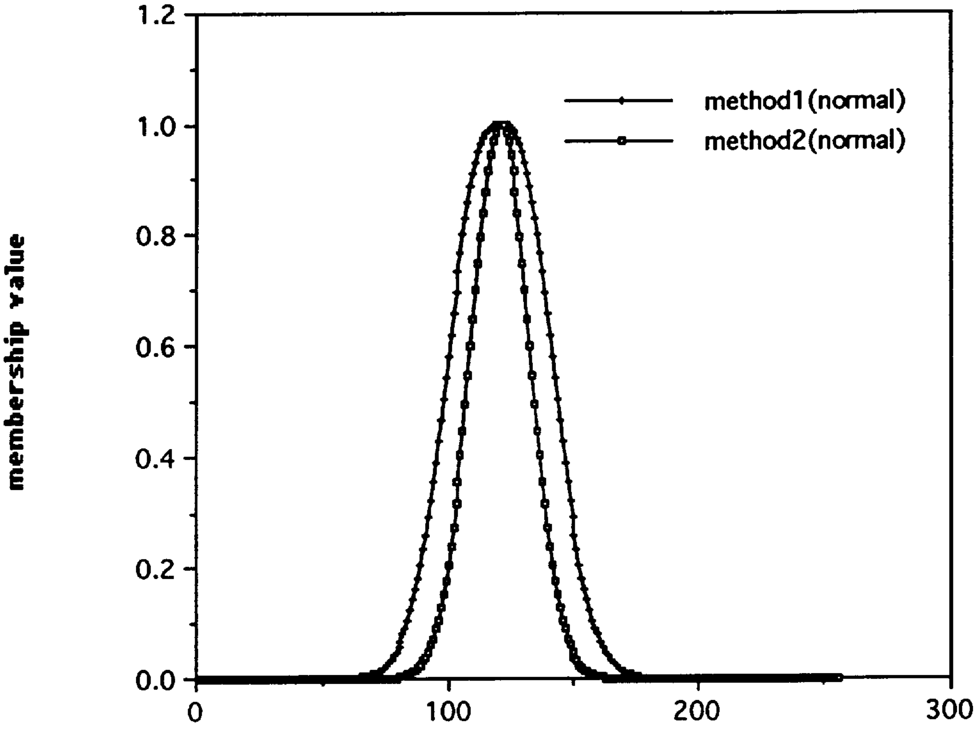
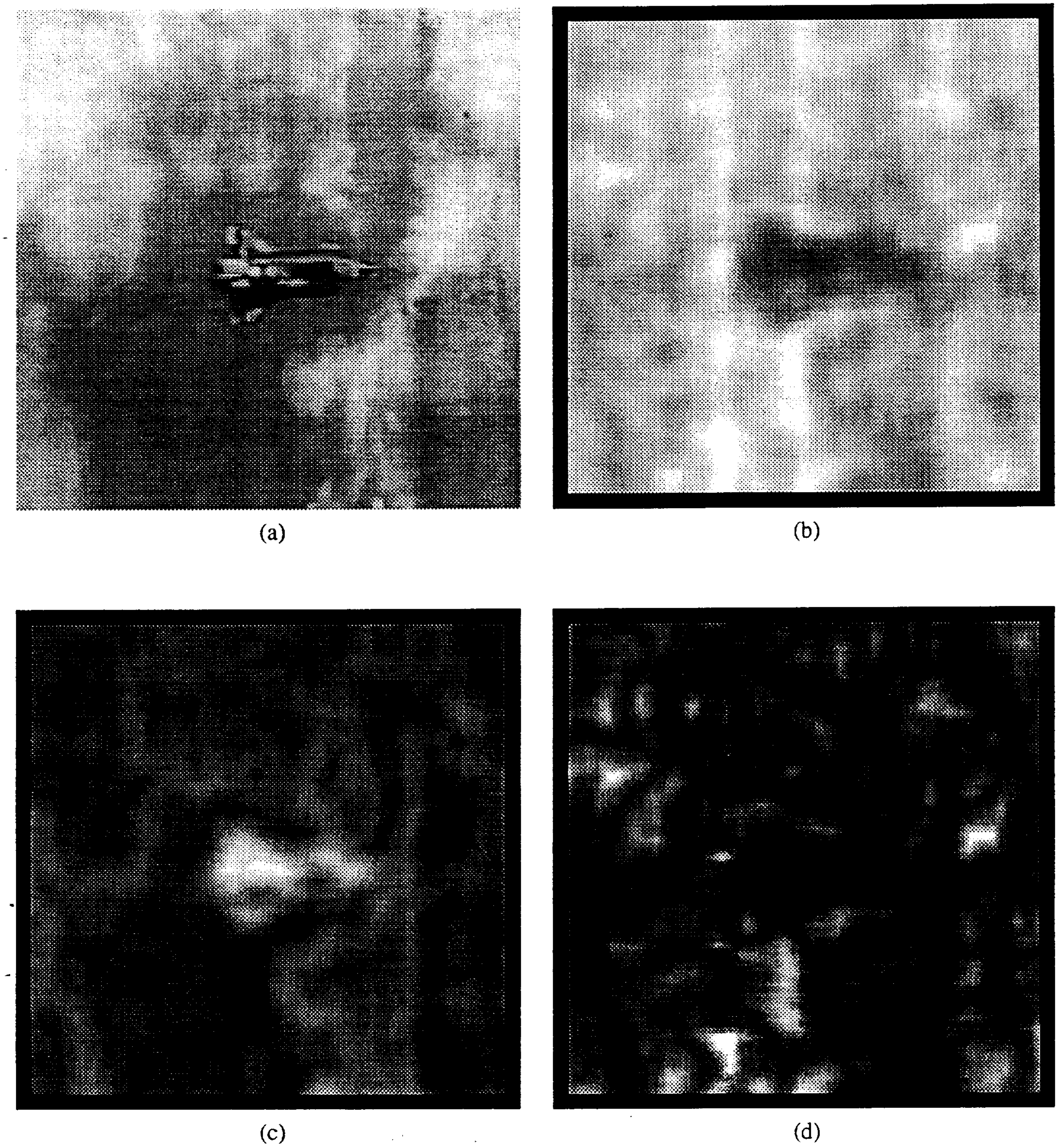


Figure 1-1 : (d') membership value of (d).



(a)

(b)

(c)

(d)

(a) original image
(c) entropy

(b) homogeneity
(d) angular second moment

Figure 2-1

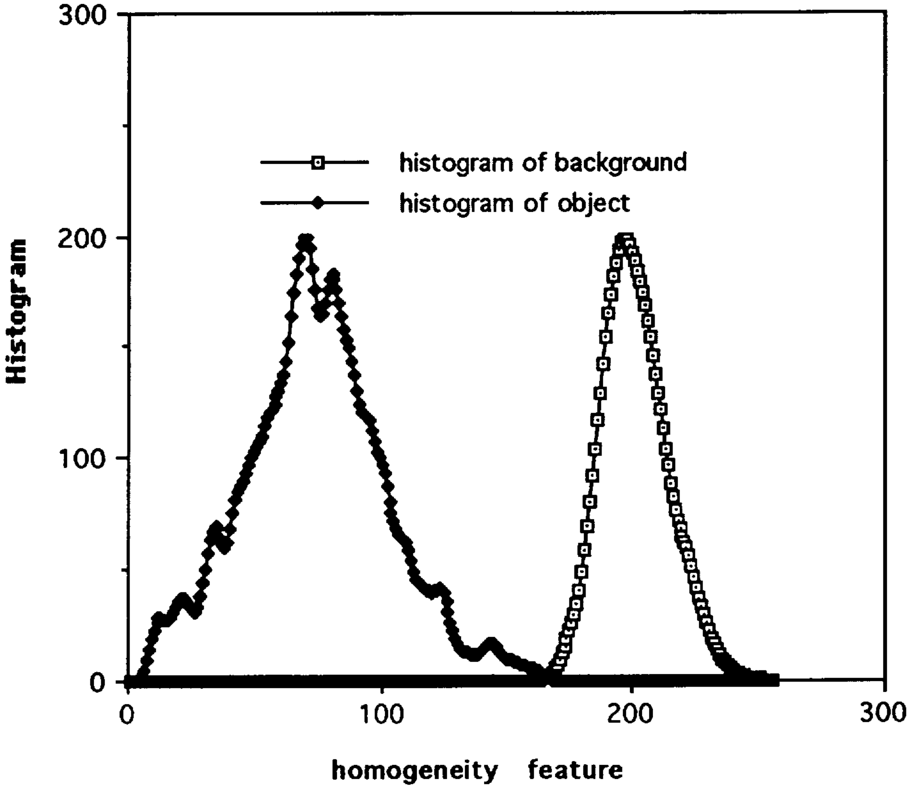


Figure 2-2 : (a) histograms of object and background.

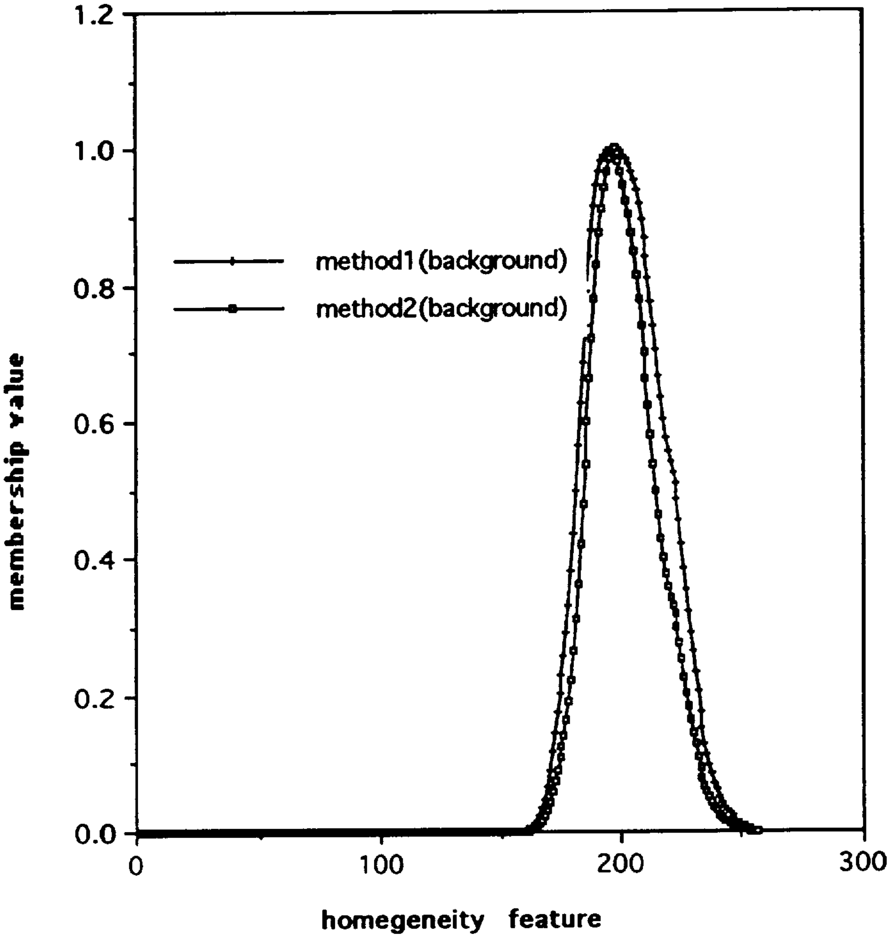


Figure 2-2 : (b) membership value of background.

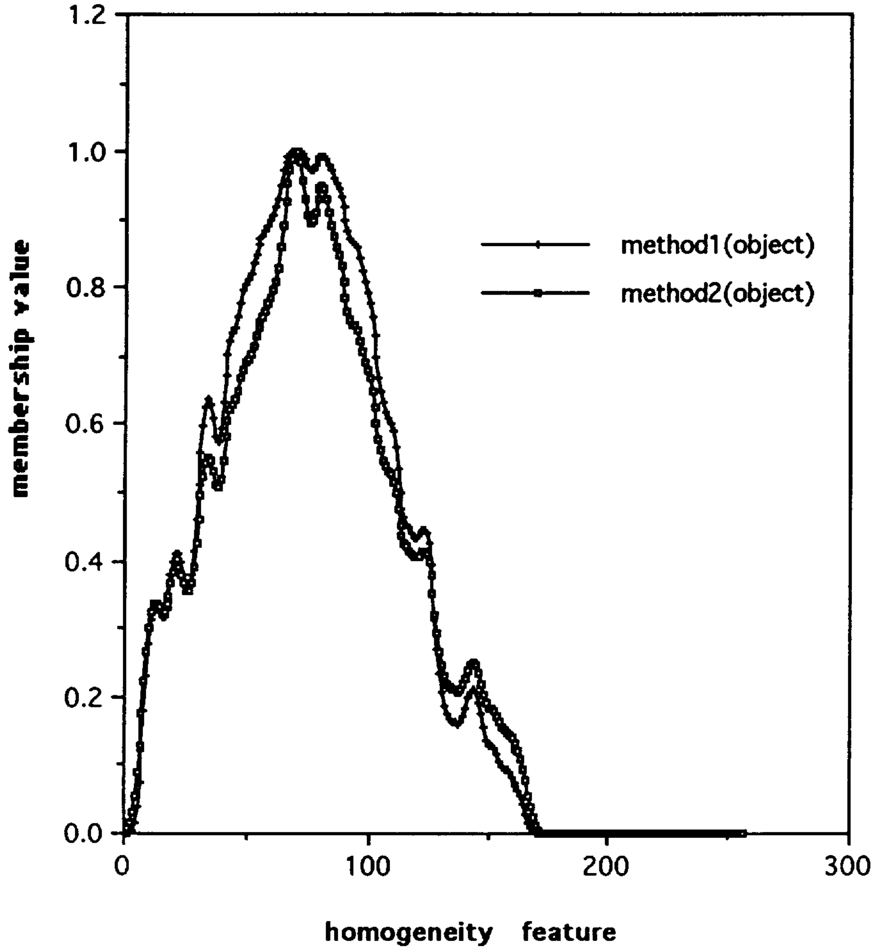


Figure 2-2 : (c) membership value of object.

Image Segmentation via Binary Encoding

1. Introduction

In this proposal, a binary encoding method is introduced for use in image processing. In particular, the area of image segmentation. Given a 2^n gray scaled image that consists of objects and background, each pixel value can be encoded by n bits that consists of gray levels(eg., for an image that has 256 gray levels, 8 bits are needed) representing the objects and background. Based on this representation, a boolean function f consisting of minimum sums-of-products(MSP) can be obtained by means of the Quine-McCluskey tabular method to segment the objects from the background. Section 2 explains how this is obtained in greater detail. The performance of this method is tested on various images, which is shown in Section 3.

2. Binary encoding method

Given n binary valued inputs(x_1,...,x_n), there exits 2^n combinations of the input variables. If we consider a binary output to represent the combinations of the input variables, a minimum sums-of-products(MSP) function f can be obtained that describes the output by means of he Quine-McCluskey tabular method. For purposes of image segmentation, given a 2^n gray scaled image that consists of objects and background, the gray levels can be encoded into n bits. If we consider the n bits as n input variables and the output to be either objects or background, a function f that segments objects from the background can be obtained by using the method described above. The coded inputs are obtained shown by the table below.

gray level	input variables $x_1 \cdots x_n$
0	$0 \cdots 0\,0$
1	$0 \cdots 0\,1$
.	.
.	.
.	.
2^n	$1 \cdots 1\,1$

The following Section shows various examples using the binary encoding method with resulting segmented images.

3. Examples

Figure 1(a) shows a 200x200 size image of the space shuttle in a background of clouds. This image consists of 256 gray levels, constituting to 8 bit encoding. Our objective is to segment the shuttle from its background using the binary encoding method described in Section 2. For the training data, a window of size 20x20 is used to take coded gray values from both the shuttle and background, constituting to 10% of the total image. For the gray values that do not exist in either the shuttle or background, are considered as "don't cares". For the gray values that are the same for both shuttle and background, they are also considered as "don't cares". Letting the shuttle and background represent the output value 1 and 0 respectively, the function f that segments the object from the background resulted as follows:

$$f(x_1,\ldots,x_n) = \bar{x}_1\bar{x}_2 + x_2x_3\bar{x}_4 + \bar{x}_1\bar{x}_3 + x_4\bar{x}_5\bar{x}_6\bar{x}_7\bar{x}_8 + \bar{x}_3x_5x_6x_7\bar{x}_8 + x_2\bar{x}_3\bar{x}_6x_7$$
$$+ x_2\bar{x}_3x_6\bar{x}_7 + x_2\bar{x}_3x_8 + x_1x_4\bar{x}_5x_6 + x_1x_5\bar{x}_6\bar{x}_7\bar{x}_8 + x_1x_4x_5x_7 .$$

The remaining 90% of the image was used for testing and the resulting segmented image is shown in Figure 1(b) using the function f above. As expected, the results showed a poor segmentation of the image. This is due to the training data which consisted of several gray levels that were the same for both the shuttle and background. This problem can also arise in many other segmentation methods. In order to improve the segmentation, several features are calculated. Figure 1(c) shows the image of the homogeneity feature and the function f that segments the object from the background using this image resulted as follows:

$$f(x_1,\ldots,x_n) = \bar{x}_1 + \bar{x}_2\bar{x}_3\bar{x}_4\bar{x}_5x_6 + \bar{x}_2\bar{x}_3x_5\bar{x}_6\bar{x}_8 .$$

Figure 1(d) shows a great improvement in the results using the homogeneity feature. Two other features, namely, entropy and contrast, were calculated and their respective images are shown in Figure 2(a) and Figure 2(c). The functions that segments the object from the background using each corresponding image resulted as follows:

For the entropy feature,

$$f(x_1,\ldots,x_n) = x_1x_4\bar{x}_7x_8 + x_1x_4x_6x_7\bar{x}_8 + x_1x_3 + x_1x_2$$

and for the contrast feature,

$$f(x_1,\ldots,x_n) = x_1 + x_2 + x_3 .$$

Figure 2(b) and Figure 2(d) show the segmented images obtained from the entropy and contrast feature images, respectively. These results also show a great improvement over the segmentation using the original image.

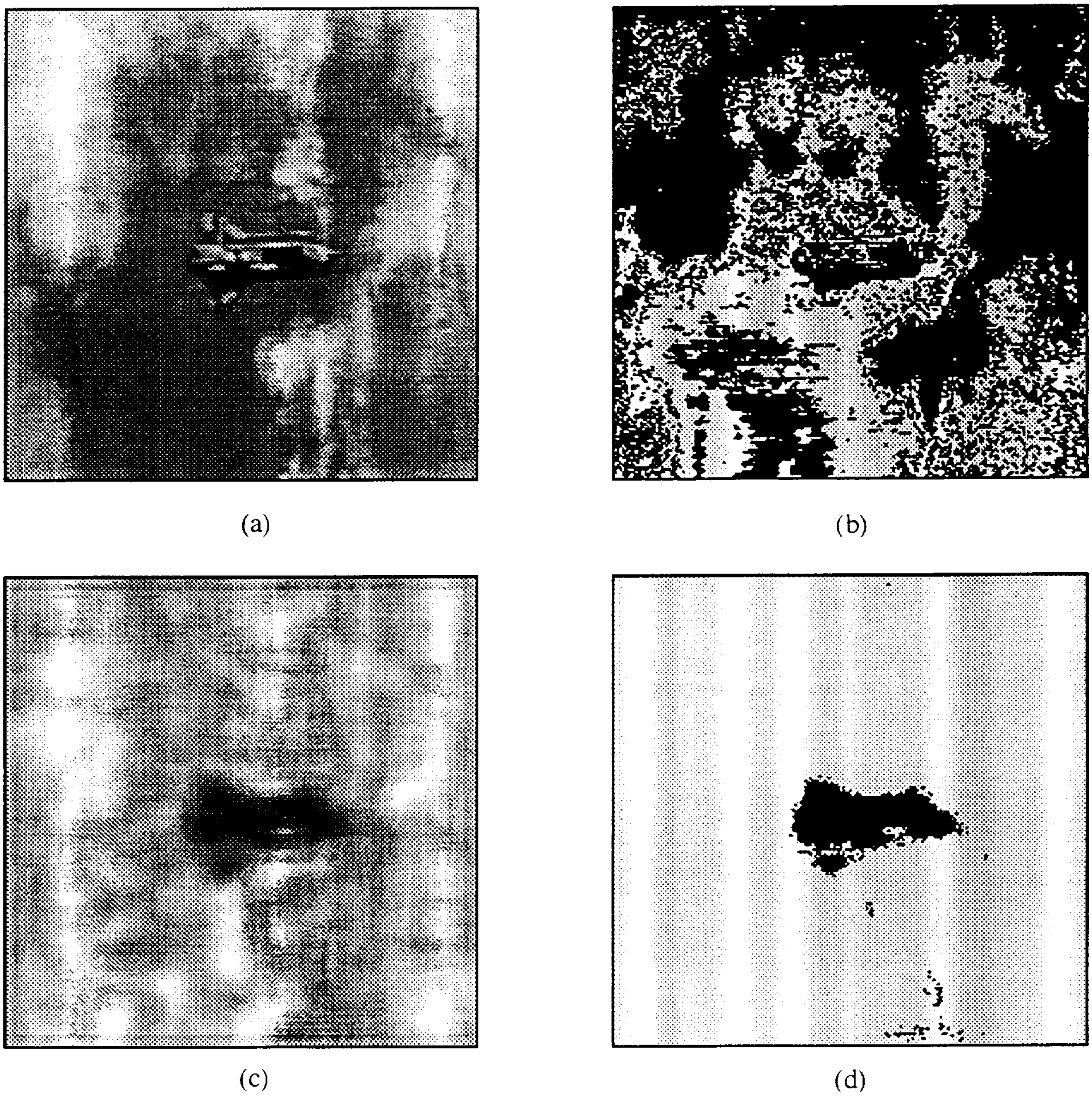
4. Bibliography

1. S. E. Hampson and D. Volper, "Representing and Learning Boolean Functions of Multi- valued Features", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 1, Jan/Feb 1990, pp. 67-79.

2. *Proceedings of the 21st International Symposium on Multiple-Valued Logic*, IEEE Computer Society Press, 1991.

3. Z. Kohavi, *Switching and Finite Automata.* New York: McGraw-Hill,1976.

4. E. J. McCluskey, *Logic Design Principles.* Englewood Cliffs, NJ: Prentice Hall, 1986.



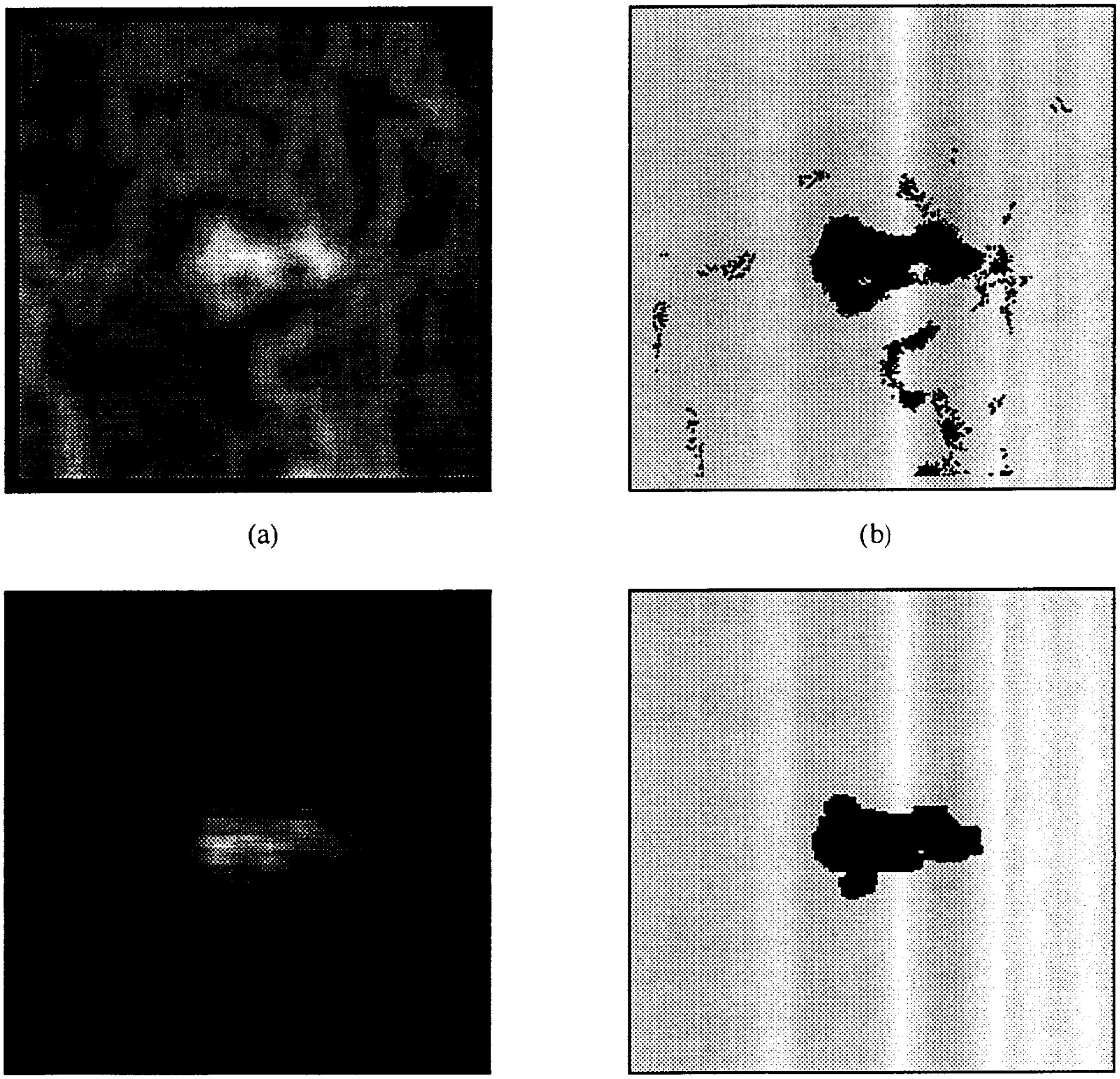
(a)

(b)

(c)

(d)

Figure 1



(a)

(b)

(c)

(d)

Figure 2

Clustering Methodologies

The best way to describe the new work in this task is to include a copy of a manuscript recently submitted by Dr. Krishnapuram and Dr. Keller to the *IEEE Transactions on Fuzzy Systems.*

The title of the paper is:

"A Possibilistic Approach to Clustering"

This represents a radical new approach to the theory and practice of fuzzy clustering.

A Possibilistic Approach to Clustering

Raghu Krishnapuram and James M. Keller
Department of Electrical and Computer Engineering
University of Missouri, Columbia, MO 65211

Abstract

Clustering methods have been used extensively in computer vision and pattern recognition. Fuzzy clustering has been shown to be advantageous over crisp (or traditional) clustering in that total commitment of a vector to a given class is not required at each iteration. Recently fuzzy clustering methods have shown spectacular ability to detect not only hypervolume clusters, but also clusters which are actually "thin shells", i.e., curves and surfaces. Most analytic fuzzy clustering approaches are derived from Bezdek's Fuzzy C-Means (FCM) algorithm. The FCM uses the probabilistic constraint that the memberships of a data point across classes sum to one. This constraint was used to generate the membership update equations for an iterative algorithm. Unfortunately, the memberships resulting from FCM and its derivatives do not correspond to the intuitive concept of degree of belonging, and moreover, the algorithms have considerable trouble in noisy environments. In this paper, we cast the clustering problem into the framework of possibility theory. Our approach is radically different from the existing clustering methods in that the resulting partition of the data can be interpreted as a possibilistic partition, and the membership values may be interpreted as degrees of possibility of the points belonging to the classes. We construct an appropriate objective function whose minimum will characterize a good possibilistic partition of the data, and we derive the membership and prototype update equations from necessary conditions for minimization of our criterion function. We illustrate the superiority of the resulting family of possibilistic algorithms (particularly in the presence of noise) with several examples.

I. Introduction

Clustering has long been a popular approach to unsupervised pattern recognition [1]. It has become more attractive with the connection to neural networks [2,3,4], and with the increased attention to fuzzy clustering [5,6,7,8]. In fact, recent advances in fuzzy clustering have shown spectacular ability to detect not only hypervolume clusters, but also clusters which are actually "thin shells", i.e., curves and surfaces [8,13]. One of the major factors that influences the determination of appropriate groups of points is the "distance measure" chosen for the problem at hand. Fuzzy clustering has been shown to be advantageous over crisp (or traditional) clustering in that total commitment of a vector to a given class is not required at each iteration.

Most analytic fuzzy clustering approaches are derived from Bezdek's Fuzzy C-Means (FCM) algorithm [14]. The FCM uses the probabilistic constraint that the memberships of a data point across classes must sum to one. This constraint came from generalizing a crisp C-Partition of a data set, and was used to generate the membership update equations for an iterative algorithm. These equations emerge as necessary conditions for a global minimum of a least-squares type of criterion function. Unfortunately, the resulting memberships do not represent one's intuitive notion of degrees of belonging, i. e., they do not represent degrees of "typicality" or "possibility".

The following simple examples illustrate the problems associated with the probabilistic constraint as related to clustering. Consider the two clusters shown in Figure 1(a) with two outlying points A and B. Intuitively point A, being an outlier, should not have a high degree of membership in either cluster. Point B should have an even smaller membership in either cluster, because it only vaguely represents either one of them. Yet, the FCM assigns a membership of 0.5 in the two clusters to both of them. Thus, the membership values are not only unrepresentative of the degree of belonging, but they cannot distinguish between a moderate outlier and an extreme outlier. Figure 1(b) represents another situation where there are two intersecting clusters. Here again, the probabilistic constraint in the FCM updates would force a membership of 0.5 in the two clusters to both point A and point B. This is again counterintuitive since point A is a "good" member of both clusters, whereas point B is a "poor" member. These situations arise because the probabilistic memberships cannot distinguish between "equal evidence" and "ignorance". More recent theories such as belief theory [15] and possibility theory [16,17] have tried to correct this problem. Zadeh suggested that membership functions of fuzzy sets can be interpreted as possibility distributions [18].

There is another important motivation for using possibilistic memberships. Like all unsupervised techniques, clustering (crisp or fuzzy) suffers from the presence of noise in the data. Since most distance functions are geometric in nature, noise points, which are often quite distant from the primary clusters, can drastically influence the estimates of the class prototypes, and hence, the final clustering. Fuzzy methods ameliorate this problem when the number of classes is greater than one, since the noise points tend to have somewhat smaller membership values in all the classes. However, this difficulty still remains in the fuzzy case, since the memberships of unrepresentative (or noise) points can still be significantly high. In fact, if there is only one real cluster present in the data, there is essentially no difference between the crisp and fuzzy methods. The prototype parameters (such as the center and orientation) and properties of the cluster (such as hypervolume) can be greatly affected by the noise in the data. Recently, Dave has suggested a heuristic method to improve the performance of the FCM algorithm and its derivatives in the presence of noise by including a noise cluster [19]. Although the results shown are good, this method introduces an artificial class and still suffers from the drawbacks due to the probabilistic constraint. His algorithm, for example, would assign a membership of about 0.5 in both classes to point A in Figure 1(b).

On the other hand, if a set of feature vectors is thought of as the domain of discourse for a collection of independent fuzzy subsets, then there should be no constraint on the sum of the memberships. The only real constraint is that the assignments do really represent fuzzy membership values, i.e., they must lie in the interval [0,1]. In this paper we cast the clustering problem into the framework of possibility theory. Our approach is fundamentally different from the existing clustering methods in that the resulting partition of the data can be interpreted as a possibilistic partition, and the membership values may be interpreted as possibility values, or degrees of typicality of the points in the classes. This is more in keeping with the concept of membership functions in fuzzy set theory. The possibilistic C-partition defines C distinct (uncoupled) possibility distributions (and the corresponding fuzzy sets) over the universe of discourse of the set of feature points. Thus, our approach is intrinsically fuzzy, in the sense that the memberships are not "hard" even when there is only one class in the data set. In section II, we construct an appropriate objective function whose minimum will characterize a good possibilistic partition of the data, and we derive the membership and prototype update equations from necessary conditions for minimization of our criterion function. These equations lead to an entirely new family of possibilistic clustering algorithms. In section III, we illustrate the superiority of the resulting family of possibilistic algorithms (particularly in the presence of noise) with several examples. Finally, section IV gives the summary and conclusions.

II. Possibilistic Clustering Algorithms

The original FCM formulation minimizes the objective function given by

$$J(L,U) = \sum_{i=1}^{C} \sum_{j=1}^{N} (\mu_{ij})^m d_{ij}^2 \text{, subject to } \sum_{i=1}^{C} \mu_{ij} = 1 \text{ for all } j. \qquad (1)$$

In (1), $L = (\lambda_1,...,\lambda_C)$ is a C-tuple of prototypes, d_{ij}^2 is the distance of feature point x_j to cluster λ_i, N is the total number of feature vectors, C is the number of classes, and $U = [\mu_{ij}]$ is a $C \times N$ matrix called the fuzzy C-partition matrix [14] satisfying the following conditions:

$$\mu_{ij} \in [0,1] \text{ for all } i \text{ and } j, \qquad \sum_{i=1}^{C} \mu_{ij} = 1 \text{ for all } j, \text{ and}$$

$$0 < \sum_{j=1}^{N} \mu_{ij} < N \text{ for all } i.$$

Here, μ_{ij} is the grade of membership of the feature point x_j in cluster λ_i, and $m \in [1,\infty)$ is a weighting exponent called the fuzzifier. In what follows, λ_i will also be used to denote the ith cluster, since it contains all of the parameters that define the prototype of the cluster.

Simply relaxing the constraint in (1) produces the trivial solution, i. e., the criterion function is minimized by assigning all memberships to zero. Clearly, one would like the memberships for representative feature points to be as high as possible, while unrepresentative points should have low membership in all clusters. This is an approach consistent with possibility theory [16]. The objective function which satisfies our requirements may be formulated as:

$$J_m(L,U) = \sum_{i=1}^{C} \sum_{j=1}^{N} (\mu_{ij})^m d_{ij}^2 + \sum_{i=1}^{C} \eta_i \sum_{j=1}^{N} (1-\mu_{ij})^m . \qquad (2)$$

where η_i are suitable positive numbers. The first term demands that the distances from the feature vectors to the prototypes be as low as possible, whereas the second term forces the μ_{ij} to be as large as possible, thus avoiding the trivial solution. The choice of η_i will be discussed later.

Theorem:

Suppose that $X = \{x_1, x_2, ..., x_N\}$ is a set of feature vectors, $L = (\lambda_1,...,\lambda_C)$ is a C-tuple of prototypes, d_{ij}^2 is the distance of feature point $\mathbf{x}_j$ to the cluster prototype λ_i, (i = 1, ..., C; j = 1, ..., N), and $U = [\mu_{ij}]$ is a $C \times N$ matrix of possibilistic membership values. Then U may be a global minimum for $J_m(L,U)$ only if $\mu_{ij} = \left[1 + \left(\frac{d_{ij}^2}{\eta_i} \right)^{\frac{1}{m-1}} \right]^{-1}$. The necessary conditions on the prototypes are identical to the corresponding conditions in the FCM and its derivatives.

Proof:

In order to derive the necessary conditions and the membership updating equations, we first note that the rows and columns of U are independent of each other. Hence, minimizing $J_m(L,U)$ with respect to U is equivalent to minimizing the following individual objective function with respect to each of the μ_{ij} (provided that the resulting solution lies in the interval [0,1]).

$$J_m^{ij}(\lambda_i,\mu_{ij}) = \mu_{ij}^m d_{ij}^2 + \eta_i (1-\mu_{ij})^m . \qquad (3)$$

Differentiating (3) with respect to μ_{ij} and setting it to zero leads to the equation

$$\mu_{ij} = \frac{1}{1 + \left(\frac{d_{ij}^2}{\eta_i}\right)^{\frac{1}{m-1}}} \tag{4}$$

It is obvious from (4) that μ_{ij} lies in the desired range. Since the newly added second term in the objective function is independent of the prototype parameters and the distance measure, the derivative of our new criterion function with respect to those parameters will be identical to that for the FCM or the appropriate generalization. QED

Thus, in each iteration, the updated value of μ_{ij} depends only on the distance of x_j. from λ_i, which is an intuitively pleasing result. The membership of a point in a cluster should be determined solely by how far it is from the prototype of the class, and should not be coupled to its location with respect to other classes. The updating of the prototypes depends on the distance measure chosen, and will proceed exactly the same way as in the case of the FCM algorithm and its derivatives, as will be explained shortly.

It is apparent from (4) that the constraints satisfied by the possibilistic C-partition are

$$\mu_{ij} \in [0,1] \text{ for all } i \text{ and } j, \qquad \text{and } 0 \leq \sum_{j=1}^{N} \mu_{ij} \leq N \text{ for all } i.$$

Eq.(4) defines a possibility distribution (membership) function for cluster λ_i over the domain of discourse consisting of all feature points $\mathbf{x}_j$. We denote this distribution by Π_i. The value of m determines the fuzziness of the final possibilistic C-partition and the shape of the possibility distribution. When $m \to 1$, the membership function is hard, and when $m \to \infty$, the memberships are maximally fuzzy. A value of 2 for m (which seems to give good results in practice), yields a very simple equation for the membership updates.

The value of η_i determines the distance at which the membership value of a point in a cluster becomes 0.5 (i. e., "the 3 dB point"). Thus, it needs to be chosen depending on the desired "bandwidth" of the possibility (membership) distribution for each cluster. This value could be the same for all clusters, if all clusters are expected to be similar. In general, it is desirable that η_i

relates to the overall size and shape of cluster λ_i. Also, it is to be noted that η_i determines the relative degree to which the second term in the objective function is important compared to the first. If the two terms are to be weighted roughly equally, then η_i should be of the order of $d^2{}_{ij}$. In practice we find that the following definition works best.

$$\eta_i = \frac{\sum_{j=1}^{N} \mu_{ij}^{m}\, d_{ij}^{2}}{\sum_{j=1}^{N} \mu_{ij}^{m}} \qquad (5)$$

This choice makes η_i the average fuzzy intra-cluster distance of cluster λ_i. The following rule may also be used.

$$\eta_i = \frac{\sum_{x_j \in (\Pi_i)_\alpha} d_{ij}^{2}}{|(\Pi_i)_\alpha|}, \qquad (6)$$

where $(\Pi_i)_\alpha$ is an appropriate α-cut of Π_i. In this case, η_i is the average intra-cluster distance for all of the "good" feature vectors (those vectors whose memberships are greater than or equal to α).

The value of η_i can be fixed for all iterations, or it may be varied in each iteration. When η_i is varied in each iteration, care must be exercised, since it may lead to instabilities. Our experience shows that the final clustering is quite insensitive to large (an order of magnitude) variations in the values of η_i, although the final shapes of the Π_i do depend on the exact values of η_i. Thus, the best approach is to compute approximate values for the η_i based on an initial fuzzy partition using (5), and after the algorithm converges, recompute more accurate values for the η_i using (6) and run the algorithm for the second time. The second run typically converges in a couple of iterations. This is only necessary if the actual values of class memberships are required. If only the relative degree of strength is needed (for example, to generate parameters for the cluster or to produce a hard partition), then this final step can be omitted. The second pass through the algorithm with refined values for η_i allows the resultant memberships in a noisy environment to be nearly identical to those obtained in a noise-free state. Any value of α between 0.1 and 0.4 seems to yield consistent results.

We propose a family of possibilistic clustering algorithms whose general form is as follows.

```
THE POSSIBILISTIC CLUSTERING ALGORITHM:
        Fix the number of clusters $C$; fix $m$, $1 < m < \infty$;
        Set iteration counter $l = 1$;
        Initialize the possibilistic $C$-partition $U^{(0)}$
                        (using a suitable fuzzy clustering algorithm);
        Estimate $\eta_i$ using (5);
        Repeat
            Update the prototypes using $U^{(l)}$, as indicated below;
            Compute $U^{(l+1)}$ using (4);
            Increment $l$;
        Until ( $\| U^{(l-1)} - U^{(l)} \| < \varepsilon$ );
        {Reestimate $\eta_i$ using (6) and rerun the repeat loop if required}
```

The updating of the prototypes depends on the distance measure chosen. Different distance measures lead to different algorithms. If the distance is an inner product induced norm metric as in the case of the FCM algorithm, i. e., if $d_{ij}^2 = (x_j-c_i)^T A_i (x_j-c_i)$ where c_i is the center of cluster λ_i, updating of the prototype is achieved by [14]

$$c_i = \frac{\sum_{j=1}^{N} \mu_{ij}^m x_j}{\sum_{j=1}^{N} \mu_{ij}^m} . \tag{7}$$

This gives us the Possibilistic C-Means (PCM) algorithm. If the distance measure is the scaled Mahalanobis distance [6,7,20], i. e., if if $d_{ij}^2 = |F_i|^{1/n}(x_j-c_i)^T F_i^{-1}(x_j-c_i)$, where F_i is the fuzzy covariance matrix of cluster $\lambda_i = (c_i, F_i)$ then the center c_i is still updated using (7), and the fuzzy covariance matrix is updated using

$$F_i = \frac{\sum_{j=1}^{N} \mu_{ij}^{m} (x_j - c_i)(x_j - c_i)^{\mathrm{T}}}{\sum_{j=1}^{N} \mu_{ij}^{m}}. \tag{8}$$

This gives us the Possibilistic C-Planes (PCP) algorithm. In the case of spherical shell clusters [9,13], one possible distance measure is $d_{ij}^2 = d^2(x_j,\lambda_i) = (\|x_j - c_i\|^2 - r_i^2)^2$, where c_i is the center and r_i is the radius of cluster λ_i, and the updating of the prototypes is given by

$$p_i = -\frac{1}{2} (H_i)^{-1} w_i, \tag{9-a}$$

where

$$p_i = \begin{bmatrix} -2\, c_i \\ c_i^{\mathrm{T}} c_i - r_i^2 \end{bmatrix}, H_i = \sum_{j=1}^{N} \mu_{ij}^{m} \begin{bmatrix} x_j \\ 1 \end{bmatrix} \begin{bmatrix} x_j \\ 1 \end{bmatrix}^{\mathrm{T}}, \text{ and } w_i = 2 \sum_{j=1}^{N} \mu_{ij}^{m} \,(x_j^{\mathrm{T}} x_j)\begin{bmatrix} x_j \\ 1 \end{bmatrix}. \tag{9-b}$$

The resulting algorithm may be called the Possibilistic C-Spherical Shells (PCSS) algorithm. A Possibilistic C-Quadric Shells (PCQS) algorithm [12,21] may also be defined similarly.

III. Examples of Possibilistic Clustering

In this section, we show several examples of possibilistic clustering to illustrate the ideas presented in the previous section. We first present a simple example to provide insights into the possibilistic approach. We then present more realistic examples, and compare the performance of the possibilistic clustering with those of the corresponding hard algorithms, and those of FCM and its derivatives.

The first example involves two well-separated clusters of seven points each. In this case, the hard C-Means algorithm, the FCM algorithm, and the PCM algorithm all give the same final crisp partition shown in Figure 2(a). The crisp partition for the FCM and PCM are obtained by assigning each feature vector to the cluster in which it has the highest membership. Ties are broken arbitrarily. The cluster centers in all three cases are the same. The membership values for the FCM and PCM cases are shown in Table 1. The feature vectors are numbered in the order in which they would be encountered in a top to bottom, left to right scan of the image shown in Figure 2(a). It can be seen that the FCM memberships are almost hard (i. e., they are close to 1 or 0) in every case. This may be desirable if a hard partition is required, but the memberships do not differentiate between close and far members of the clusters. On the other hand, the PCM algorithm provides more graded membership values, and these membership values are more in keeping with one's intuitive notion of belonging. Note that the farther away the feature vector is to the typical member (i. e., the prototype), the smaller the membership. As we noted in the previous section, the rate of fall of the membership values can be adjusted depending on the choice of η_i. However, this has virtually no effect on the final clustering obtained. We chose to keep the computation of η_i the same for all examples, and this was done as explained in the previous section.

Figures 2(b), 2(c), and 2(d) show the final crisp partition obtained due to the hard *C*-Means, FCM, and PCM algorithms respectively, when two noise points are added to the set of feature vectors shown in Figure 2(a). The hard *C*-Means algorithm actually puts the farthest noise point as one cluster, and lumps all the rest into another cluster, although this may depend on initialization. The crisp partitions of the FCM and PCM are identical, however, the membership values and the cluster centers obtained are considerably different, as can be seen in Table 2. The first two entries in the table correspond to the two noise points, and the FCM algorithm gives approximately equal memberships of 0.5 in both clusters for the noise points. This significantly affects the estimates of the cluster centers, as can be seen in the table. The PCM algorithm, on the other hand, gives very low memberships for the two noise points in either cluster, and the farther point has a lesser membership than the closer one, as desired. As a result, the cluster centers are virtually unchanged. The membership values of the points in each of the clusters is also virtually unchanged in spite of the addition of the noise points. In fact, the memberships will not change even if an entire new cluster of feature points is added to the data set. This is a highly desirable result, especially if the clustering algorithm is to be used to estimate membership distribution functions for the various classes.

Figure 3 shows a more realistic example with two classes. Each class has 25 feature vectors in the noise-free case. These points were generated with a Gaussian random number generator. Figure 3(a) shows the clustering obtained by the Hard *C*-Means, Fuzzy *C*-Means, and Possibilistic *C*-Means algorithms. The crisp partitions are identical. Figures 3(b), 3(c) and 3(d) show the crisp partition resulting from the Hard *C*-Means, Fuzzy *C*-Means, and Possibilistic *C*-Means algorithms when the data from class 2 (the lower class) are noisy. The crisp partition due to the Hard *C*-Means is quite miserable, and the crisp partition due to the Fuzzy *C*-Means is not satisfactory either. The performance of the possibilistic *C*-Means is quite acceptable. The cluster centers for the three methods for the noise-free and noisy cases are shown in Table 3. As can be seen, the center estimates are poor in the cases of the Hard and Fuzzy *C*-Means algorithms.

Figure 4 shows an example involving linear clusters. Figure 4(a) shows the clustering due to the Hard C Planes (HCP) [22,23], Fuzzy C Planes(FCP) [6,7], and Possibilistic C Planes (PCP) algorithms. The final estimates of the prototypes are shown superimposed on the original data set. The results are identical when there is no noise. Figure 4(b), 4(c), and 4(d) show the results of the HCP, FCP, and PCP algorithms respectively, when noise is added. As can be seen, the results of both the HCP and the FCP algorithms are quite poor. However, the results of the PCSS algorithm are virtually the same as those of noise-free case.

The fourth example involves the detection of circles. Figure 5(a) shows the original data set with two circles. Figure 5(b), 5(c), and 5(d) show the results of the Hard C Spherical Shells HCSS [13], Fuzzy C Spherical Shells (FCSS) [9,13], and Possibilistic C Spherical Shells (PCSS) algorithms respectively, with the final estimates of the prototypes superimposed on the original data set. As can be seen, the results of the HCSS algorithm are very poor. (They correspond to a local minimum). The FCSS and PCSS algorithms give the same results in this case. Figure 6 shows the results of the same data set when noise is added. The performance of the HCSS algorithm is again unacceptable. The FCSS algorithm performs better, however, the estimates of the centers and the radii suffer from the presence of noise. The results of the PCSS algorithm are virtually unaffected by noise.

The greatest difference between the FCM-based and PCM-based algorithms is for the case where there is but one cluster in the data set. In this case there is essentially no difference between the FCM-based methods and hard methods. Figure 7 illustrates this idea. Figure 7(a) and 7(c) show the estimates of the prototype parameters for a noisy line and a noisy circle when the FCP and FCSS algorithms are used. The estimates are severely affected by noise. Figures 7(b) and 7(d) show the clearly superior estimates with the PCP and PCSS algorithms.

IV. Conclusions

In this paper, we present a possibilistic approach to objective-function-based clustering. We argue that the existing fuzzy clustering methods do not provide intuitively appealing membership values due to the fact that an inherently probabilistic constraint is used. As a result, membership of a feature vector in a cluster depends not only on where the feature vector is located with respect to the cluster, but also on how far away it is with respect to other clusters. This "conservation of total membership" law forces the memberships to be spread across the classes, and thus makes them dependent on the number of clusters present. The resulting membership values cannot always distinguish between good members and poor members. This situation arises because probabilistic membership values cannot distinguish between "equally likely" and "unknown". On the other hand, if one takes the possibilistic view that the membership of a feature vector in a class has nothing to do with its membership in other classes, then we can achieve more realistic membership distributions. Our possibilistic approach to clustering is based on this idea.

Since our membership functions correspond more closely to the notion of typicality, the resulting algorithms are naturally more immune to noise. Thus, our approach is intrinsically fuzzy, since the memberships are not "hard" even when there is only one class in the data set. This is compatible with the fuzzy set theoretic notion of membership functions. The partition of the data resulting from our approach can be interpreted as a possibilistic partition, and the membership values may be interpreted as possibility values, or degrees of typicality of the points in the classes. The possibilistic C-partition defines C distinct (uncoupled) possibility distributions (and the corresponding fuzzy sets) over the universe of discourse of the set of feature points. Therefore, the family of algorithms we propose can be used to estimate possibility distributions directly from training data. Currently there are no good algorithms to estimate possibility distributions directly from training data, other than those that do so by converting probabilities to possibilities [24,25]. This conversion does not yield very appropriate results when the FCM-based memberships are used, since the memberships do not have a frequency interpretation, and since the memberships have already lost the distinction between "equally highly likely" and "equally highly unlikely". The possibilistic approach has the added advantage of being a natural mechanism to assign "fuzzy labels" to training data for use in more sophisticated pattern recognition algorithms. Finally, we would like to point out that the possibilistic algorithms may be viewed as a generalization of the weighted least squares approaches [26] and robust parameter estimation methods [27], which have been used with good results in computer vision [28,29].

Acknowledgment

We are grateful to our students Hichem Frigui and Olfa Nasraoui without whose suggestions and assistance the simulation experiments would not have been possible.

V. References

1. R. Dubes and A. K. Jain, *Algorithms for clustering data*, Prentice Hall, Englewood Cliffs, 1988.

2. B. Kosko, *Neural Networks and Fuzzy Systems*, Prentice Hall, Englewood Cliffs, NJ, 1992, Chapter 4.

3. T. Huntsberger and P. Ajjimarangsee, "Parallel Self-Organizing Feature Maps for unsupervised pattern recognition", International Journal of General Systems, vol. 16, pp. 357-372, 1990.

4. J. Bezdek, E.C.K. Tsao, and N. Pal, "Kohonen clustering networks", Proceedings of the First IEEE Conference on Fuzzy Systems, San Diego, March 1992.

5. I. Gath and A. Geva, "Unsupervised optimal fuzzy clustering", *IEEE Trans. PAMI* vol 11, pp. 773-781, 1989.

6. R. Krishnapuram and C.-P. Freg, "Algorithms to Detect Linear and Planar clusters and Their Applications", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, June 1991, pp. 426-431.

7. R. Krishnapuram and C.-P. Freg, "Fitting an Unknown Number of Lines and Planes to Image Data through Compatible Cluster Merging", accepted for publication in *Pattern Recognition.*

8. R. N. Dave, "New measures for evaluating fuzzy partitions induced through C-shells clustering", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, Nov. 1991, pp. 406-414.

9. R. N. Dave, "Fuzzy-shell clustering and applications to circle detection in digital images", *International Journal of General Systems*, vol. 16, 1990, pp. 343-355.

10. R. N. Dave, "Adaptive C-shells clustering", *Proceedings of the North American Fuzzy Information Processing Society Workshop*, Columbia, Missouri, 1991, pp. 195-199.

11. J. C. Bezdek and R. J. Hathaway, "Accelerating convergence of the Fuzzy C-Shells clustering algorithms", *Proceedings of the International Fuzzy Systems Association Congress*, Brussels, July 1991, Volume on *Mathematics*, pp. 12-15.

12. R. Krishnapuram, H. Frigui, and O. Nasraoui, "New fuzzy shell clustering algorithms for boundary detection and pattern recognition", *Proceedings of the SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques*, Boston, Nov. 1991, pp.458-465.

13. R. Krishnapuram, O. Nasraoui, and H. Frigui, "The fuzzy C spherical shells algorithm: A new approach", to appear in the *IEEE Transactions on Neural Networks.*

14. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.

15. G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, New Jersey, 1976.

16. D. Dubois and H. Prade, *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York, 1988.

17. G. Klir and T. Folger, *Fuzzy Sets, Uncertainty, and Information*, Prentice Hall, Englewood Cliffs, NJ, 1988, Chapter 4.

18. L. A. Zadeh, "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems*, vol. 1, 1978, pp. 3-28.

19. R. N. Dave, " Characterization and detection of noise in clustering", to appear in *Pattern Recognition Letters.*

20. D. E. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix", *Proceedings of IEEE-CDC*, vol. 2, (K. S. Fu ed.), IEEE Press, N. J., pp. 761-766, 1979.

21. R. Krishnapuram, H. Frigui, and O. Nasraoui, "Quadric shell clustering algorithms and their applications", under review.

22. E. Diday, A. Schroeder and Y. Ok, "The Dynamic Clusters Method in Pattern Recognition," *Proceedings of the International Federation for Information Processing Congress*, American Elsevier, North Holland, 1974, pp. 691-697.

23. G. S. Sebestyen, *Decision-Making Processes in Pattern Recognition*, Macmillan Company, New York, 1962.

24. D. Dubois and H. Prade, "Unfair coins and necessity measures: Towards a possibilistic interpretation of histograms", *Fuzzy Sets and Systems*, vol. 10, 1983, pp. 15-20.

25. B. Bharathi Devi and V. V. S. Sarma, "Estimation of fuzzy memberships from histograms", *Information Sciences*, vol. 35, 1985, pp. 43-59.

26. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. I, Addision Wesley, Reading, MA, 1992, Chapter 11.

27. X. Zhuang, T. Wang, and P. Zhang, "A highly robust estimator through partially likelihood function modeling and its application in computer vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 1, January 1992, pp. 19-35.

28. D. G. Lowe, "Fitting parametrized three-dimensional models to images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 5, May 1991, pp. 441-450.

29. P. Whaite and F. P. Ferrie, "From uncertainty to visual exploration", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, October 1990, pp. 1038-1049.

Table 1: Memberships and centers resulting from the FCM and PCM values for the noise-free data set shown in Figure 2(a).

	Fuzzy	*C*-Means	Possibilistic	*C*-Means
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
1	0.996	0.004	0.632	0.007
2	0.004	0.996	0.007	0.632
3	0.988	0.012	0.300	0.005
4	0.997	0.003	0.631	0.006
5	1.000	0.000	1.000	0.007
6	0.996	0.004	0.632	0.008
7	0.980	0.020	0.300	0.009
8	0.020	0.980	0.009	0.300
9	0.004	0.996	0.008	0.632
10	0.000	1.000	0.007	1.000
11	0.003	0.997	0.006	0.631
12	0.012	0.988	0.005	0.300
13	0.996	0.004	0.632	0.007
14	0.004	0.996	0.007	0.632
centers	(60.0, 150.0)	(140.0, 150.0)	(60.0, 150.0)	(140.0, 150.0)

Table 2: Memberships and centers resulting from the FCM and PCM values for the noisy data set shown in Figures 2(c) and 2(d).

	Fuzzy *C*-Means		Possibilistic *C*-Means	
	Cluster 1	Cluster 2	Cluster 1	Cluster 2
1	0.499	0.501	0.004	0.004
2	0.498	0.502	0.017	0.017
3	0.999	0.001	0.636	0.007
4	0.001	0.999	0.007	0.636
5	0.977	0.023	0.299	0.005
6	0.989	0.011	0.626	0.006
7	0.996	0.004	1.000	0.007
8	0.994	0.004	0.644	0.008
9	0.985	0.015	0.307	0.009
10	0.015	0.985	0.009	0.307
11	0.004	0.996	0.008	0.644
12	0.004	0.996	0.007	1.000
13	0.011	0.989	0.006	0.626
14	0.023	0.977	0.005	0.299
15	0.985	0.015	0.634	0.007
16	0.015	0.985	0.007	0.634
centers	(62.8,145.9)	(137.2,145.9)	(60.0,150.0)	(139.9,150.0)

Table 3: The estimates of centers using the HCM, FCM and PCM algorithms

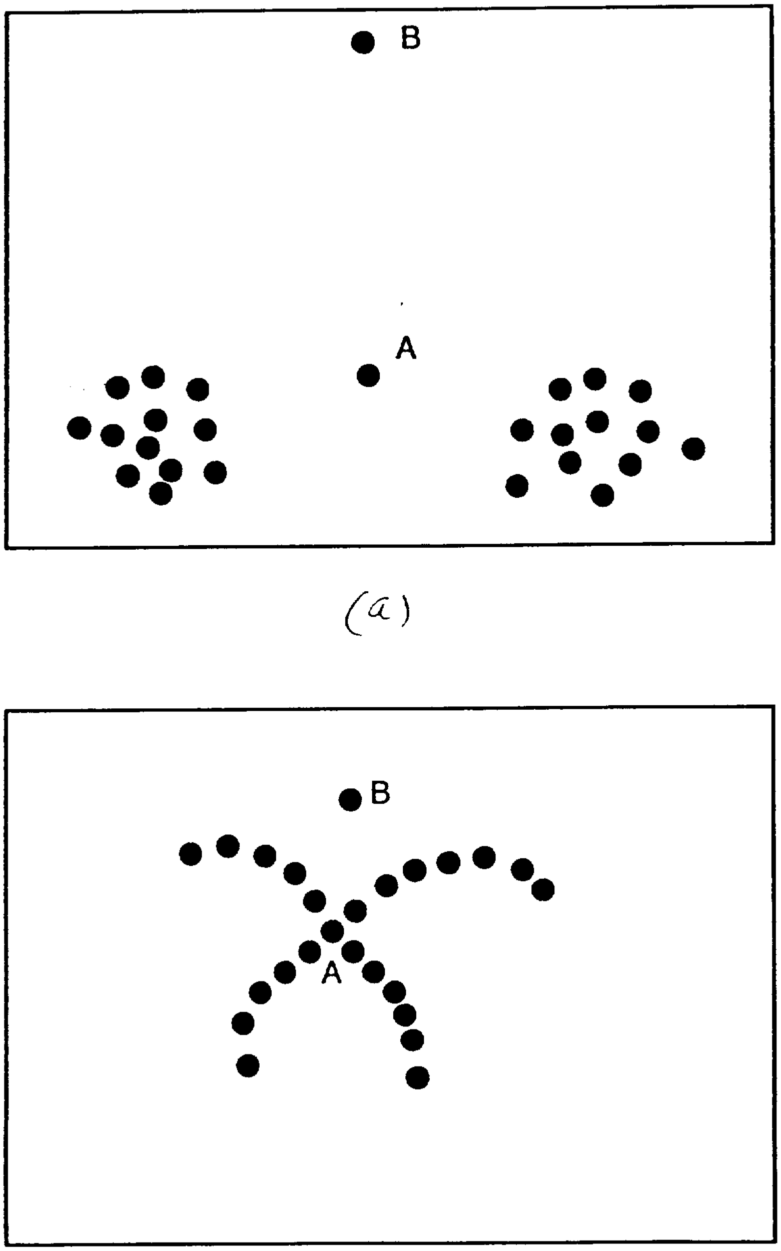
	Hard *C*-Means	Fuzzy *C*-Means	Possibilistic *C*-Means
No noise	(102.0, 88.4) (81.9, 118.5)	(102.0, 87.5) (82.3,118.1)	(101.7, 87.9) (82.4, 117.4)
With noise	(92.2, 103.8) (46.7, 155.8)	(96.4, 95.8) (66.2, 139.4)	(98.7, 93.9) (86.4, 112.6)

List of Figures

Figure 5: Estimation of parameters of circles when no noise is present (The circles generated from the estimated prototype parameters are superimposed on the original data set): (a) Original data set. (b) Parameter estimates obtained with the HCSS algorithm. (c) Parameter estimates obtained with the FCSS algorithm. (c) Parameters obtained with the PCSS algorithm.

Figure 6: Estimation of parameters of circles in noise (The circles generated from the estimated prototype parameters are superimposed on the original data set): (a) Original data set. (b) Parameter estimates obtained with the HCSS algorithm. (c) Parameter estimates obtained with the FCSS algorithm. (c) Parameters obtained with the PCSS algorithm .

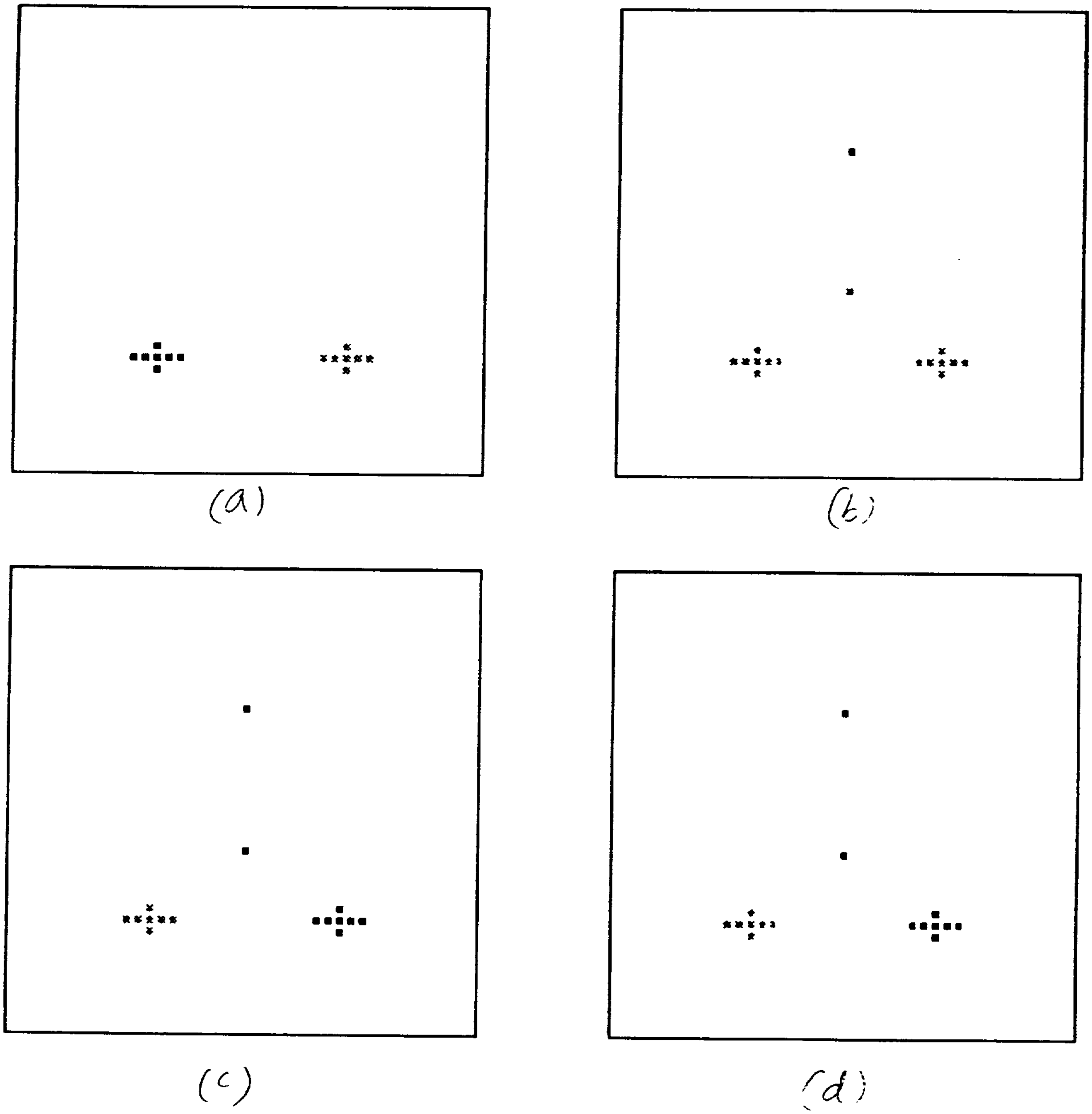
Figure 7: Estimation of prototype parameters in noise when only one cluster is present: (a) Line parameters obtained with the FCP algorithm. (b) Line parameters obtained with the PCP algorithm. (c) Circle parameters obtained with the FCSS algorithm. (d) Circle parameters obtained with the PCSS algorithm.



(a)

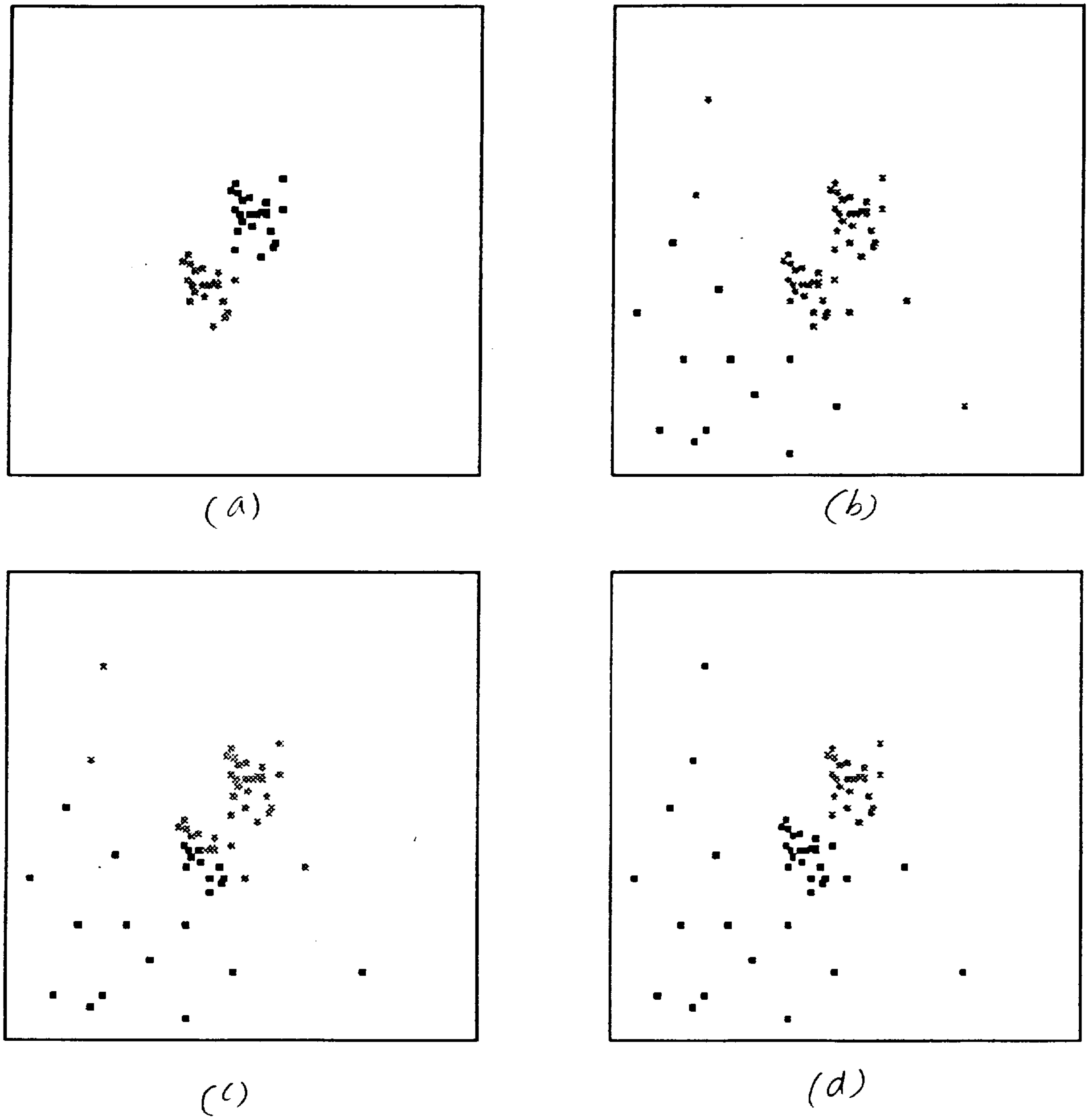
(b)

Figure 1



(a) (b) (c) (d)

Figure 2



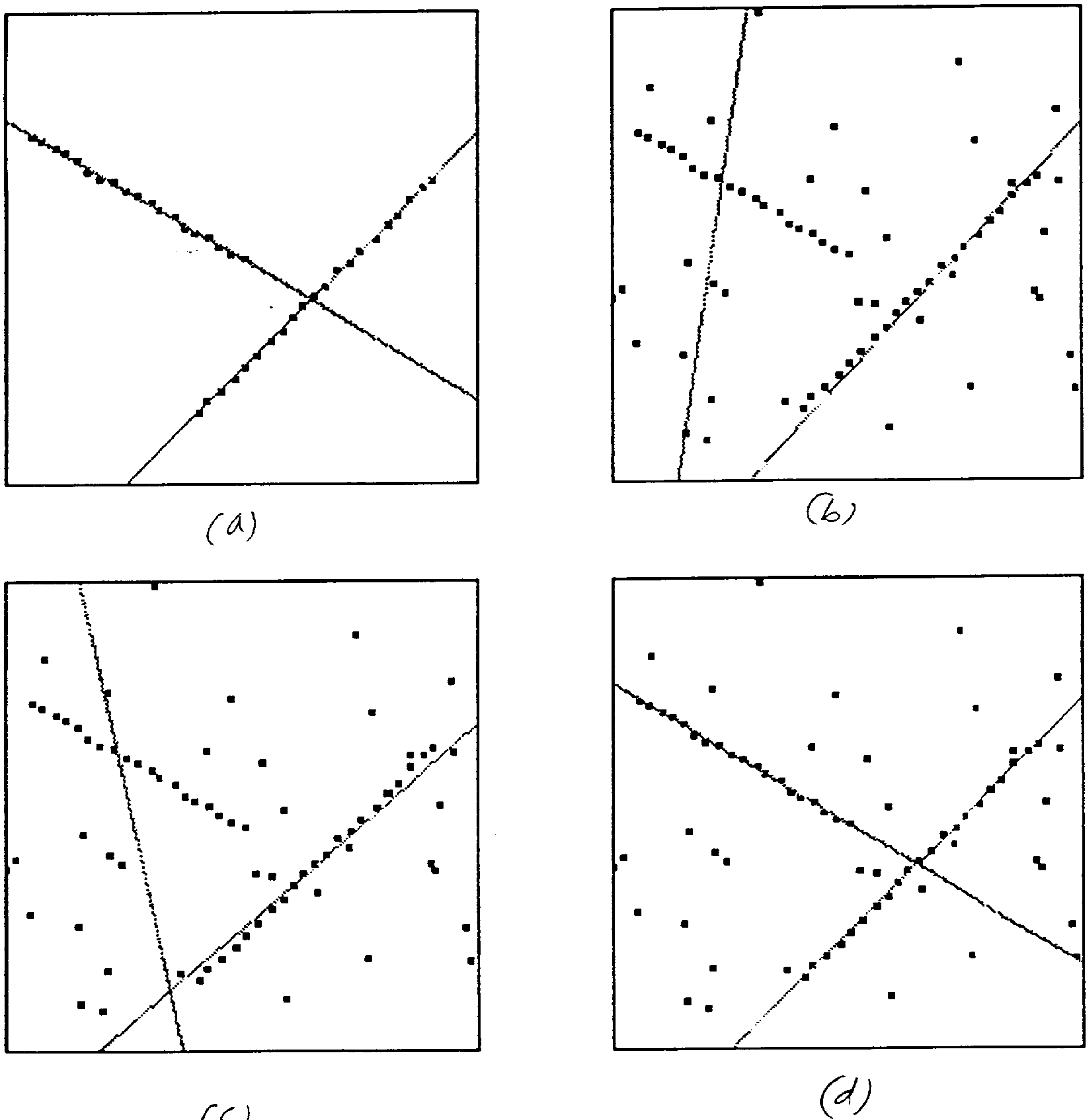
(a)

(b)

(c)

(d)

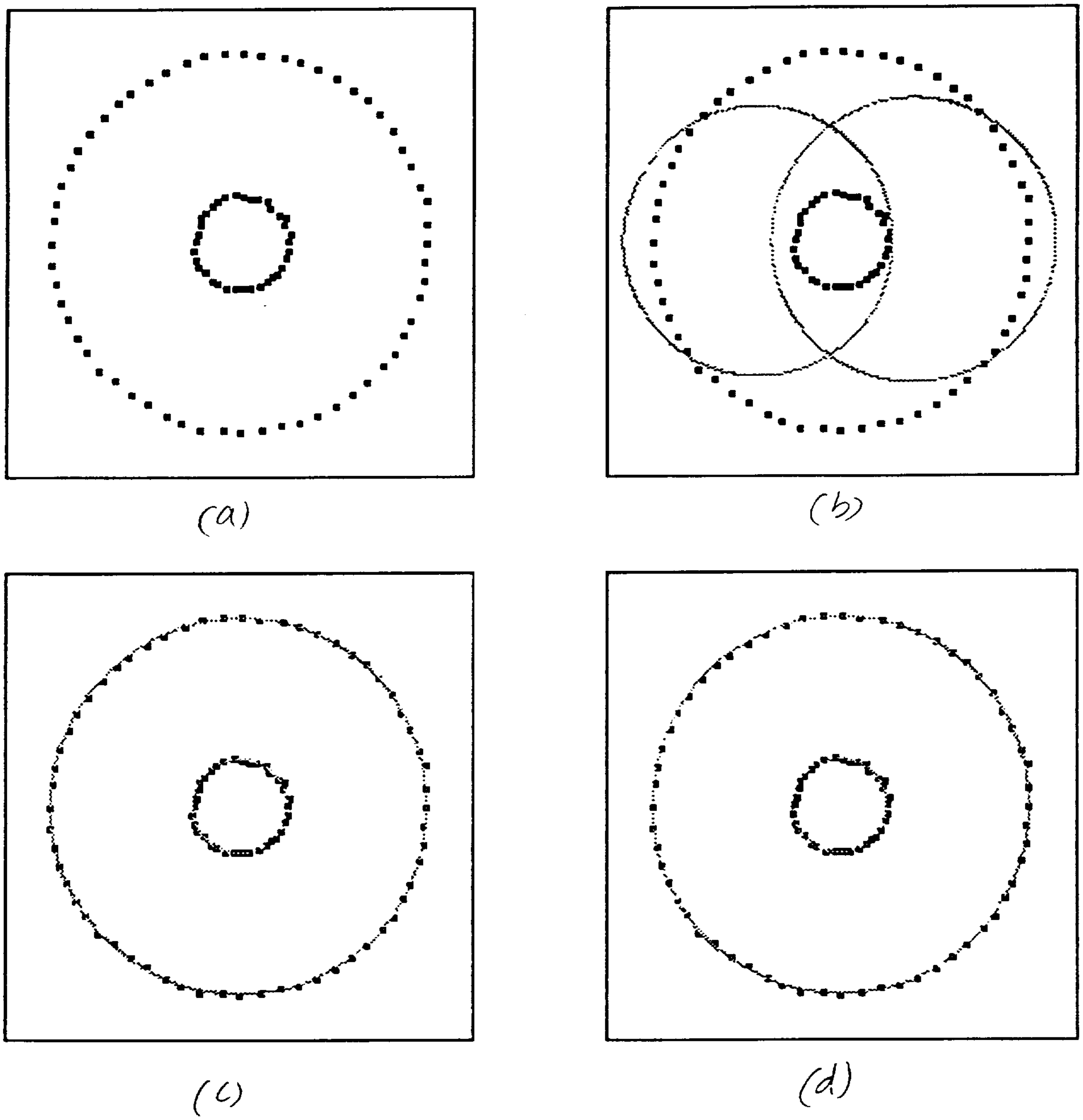
Figure 3



(a) (b)

(c) (d)

Figure 4



(a) (b) (c) (d)

Figure 5

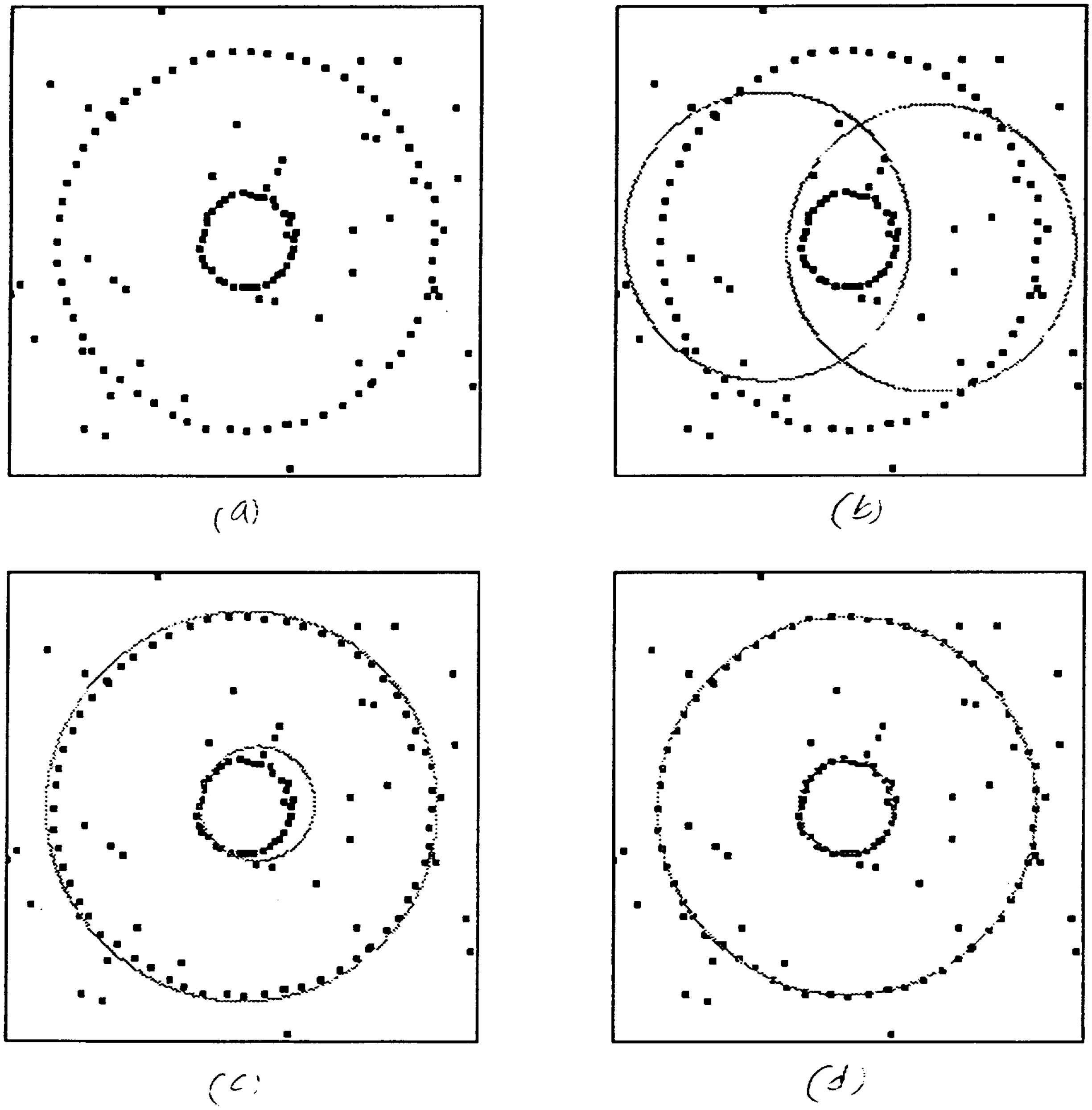
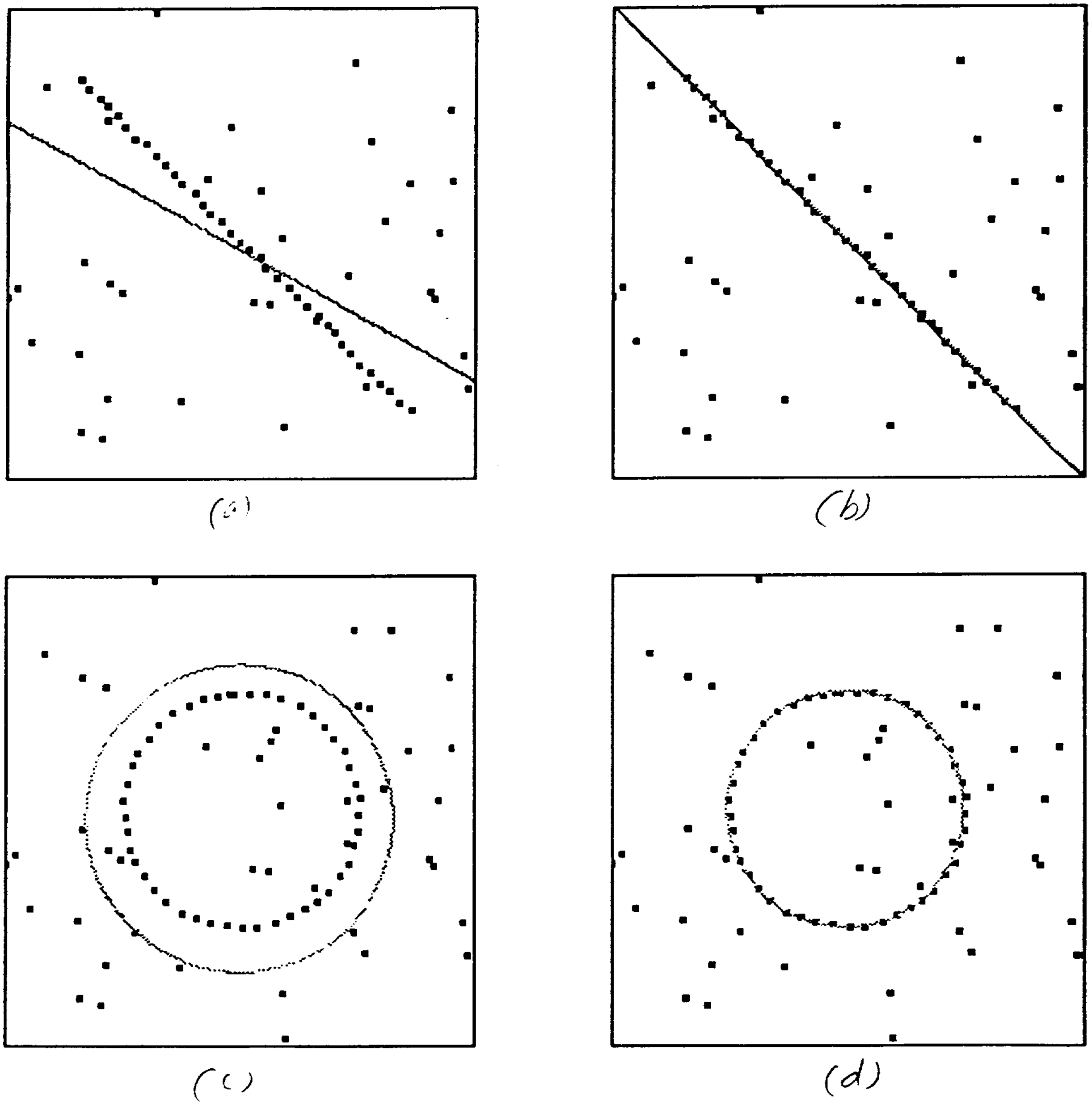


Figure 6



(a)

(b)

(c)

(d)

Figure 7

Pose Estimation Using the UCQS algorithm

In some cases, the Unsupervised C Quadric Shells (UCQS) algorithm can be used to estimate the pose of the shuttle. The shuttle's image is taken from the back so that the exhaust nozzles and the back edges of the three wings are apparent. Given an original unrotated image, the exhaust nozzles can be parametrized by three circles, and the three wings can be parametrized by three straight lines. These parameters are easily determined by the UCQS algorithm. As the shuttle rotates, the shape of the nozzles will change from circles to ellipses, so will the orientation of the straight lines representing the three wings. The UCQS algorithm is used in order to cluster this edge image and determine the parameters of the ellipses and lines. Finally, these parameters can be used to solve for the translation and rotation parameters, as long as the translation is made in the image plane. In fact, depth information can also be derived from the change in the size of the nozzles.

The (UCQS) algorithm was used to cluster edge images of the back of shuttle model. See figures C-1 through C-3. Fig.C-1 shows an image of the unrotated shuttle (the reference position), and Fig.C-2 and C-3 show images of the rotated shuttle. In all the figures, Fig.a shows the original gray level image taken of the back of the shuttle. Fig.b shows the corresponding edge image, and Fig.c shows the prototypes of the clusters found by the (UCQS) algorithm. The (UCQS) algorithm not only is able to cluster the image correctly, but it also determines the parameters of each cluster.

The equation of a quadric shell in the 2-D case (quadratic curve) can be written as follows:

$$a_1 x_1^2 + a_2 x_2^2 + a_3 x_1 x_2 + a_4 x_1 + a_5 x_2 + a_6 = 0.$$

This equation can also be written as

$$\mathbf{x}^t A\ \mathbf{x} + \mathbf{x}^t \boldsymbol{v} + d = 0.$$

where $\mathbf{x}$ is the feature vector $(x_1, x_2)^t$, and the parameters of the cluster are: A which is a 2×2 matrix, $\boldsymbol{v}$ which is 2-element vector and d which is a real number. The (UCQS) algorithm finds these parameters for figures C-1 through C-3, and they are tabulated in tables C-1 through C-3.

Table.C-1

cluster	A	v	d
1	$\begin{bmatrix}0.739 & 0.034\\ 0.034 & 0.672\end{bmatrix}$	$\begin{bmatrix}-150.8\\ -124.8\end{bmatrix}$	12517.1
2	$\begin{bmatrix}0.82 & 0.003\\ 0.003 & 0.573\end{bmatrix}$	$\begin{bmatrix}-89.3\\ -130.4\end{bmatrix}$	9454.9
3	$\begin{bmatrix}0.765 & 0.014\\ 0.014 & 0.644\end{bmatrix}$	$\begin{bmatrix}-81.0\\ -75.3\end{bmatrix}$	3878.5
4	$\begin{bmatrix}0.0 & 0.0\\ 0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}-0.006\\ 0.99\end{bmatrix}$	-89.9
5	$\begin{bmatrix}0.997 & 0.038\\ 0.038 & 0.06\end{bmatrix}$	$\begin{bmatrix}-112.5\\ -17.9\end{bmatrix}$	3238.9
6	$\begin{bmatrix}0.0 & 0.0\\ 0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}0.99\\ 0.004\end{bmatrix}$	-43.6
7	$\begin{bmatrix}0.629 & -0.246\\ -0.246 & 0.695\end{bmatrix}$	$\begin{bmatrix}-111.9\\ -10.2\end{bmatrix}$	5893.1
8	$\begin{bmatrix}0.0 & 0.0\\ 0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}0.99\\ -0.022\end{bmatrix}$	-27.6
9	$\begin{bmatrix}0.567 & 0.137\\ 0.137 & 0.801\end{bmatrix}$	$\begin{bmatrix}-148.4\\ -226.5\end{bmatrix}$	21186.7
10	$\begin{bmatrix}0.0 & 0.0\\ 0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}-0.019\\ 0.99\end{bmatrix}$	-79.5
11	$\begin{bmatrix}0.0 & 0.0\\ 0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}0.99\\ 0.004\end{bmatrix}$	-76.9

Table.C-2

cluster	A	v	d
1	$\begin{bmatrix}0.688 & 0.027\\0.027 & 0.725\end{bmatrix}$	$\begin{bmatrix}-177.2\\-149.8\end{bmatrix}$	18155.9
2	$\begin{bmatrix}0.750 & -0.058\\-0.058 & 0.656\end{bmatrix}$	$\begin{bmatrix}-122.4\\-80.9\end{bmatrix}$	7840.2
3	$\begin{bmatrix}0.759 & -0.003\\-0.003 & 0.650\end{bmatrix}$	$\begin{bmatrix}-133.9\\-160.5\end{bmatrix}$	15601.5
4	$\begin{bmatrix}0.0 & 0.0\\0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}0.99\\-0.01\end{bmatrix}$	-33.3
5	$\begin{bmatrix}0.0 & 0.0\\0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}0.004\\0.99\end{bmatrix}$	-92.7

Table.C-3

cluster	A	v	d
1	$\begin{bmatrix}0.817 & -0.065\\-0.065 & 0.569\end{bmatrix}$	$\begin{bmatrix}-187.4\\-57.9\end{bmatrix}$	12838.2
2	$\begin{bmatrix}0.806 & 0.031\\0.031 & 0.591\end{bmatrix}$	$\begin{bmatrix}-199.5\\-140.6\end{bmatrix}$	19588.9
3	$\begin{bmatrix}0.864 & 0.026\\0.026 & 0.503\end{bmatrix}$	$\begin{bmatrix}-266.5\\-99.9\end{bmatrix}$	24515.2
4	$\begin{bmatrix}0.0 & 0.0\\0.0 & 0.0\end{bmatrix}$	$\begin{bmatrix}0.99\\0.038\end{bmatrix}$	-74.3

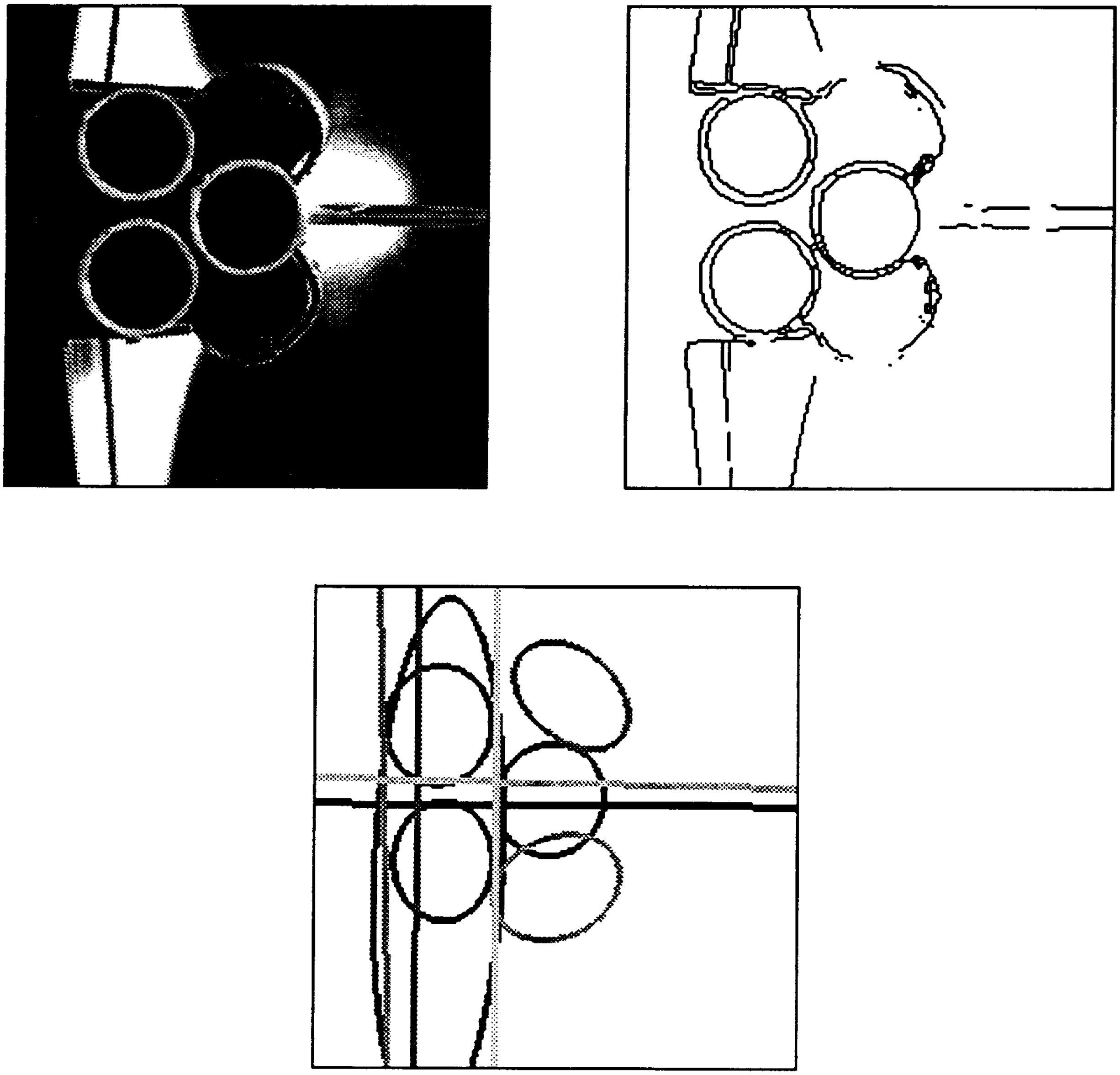


Fig 1

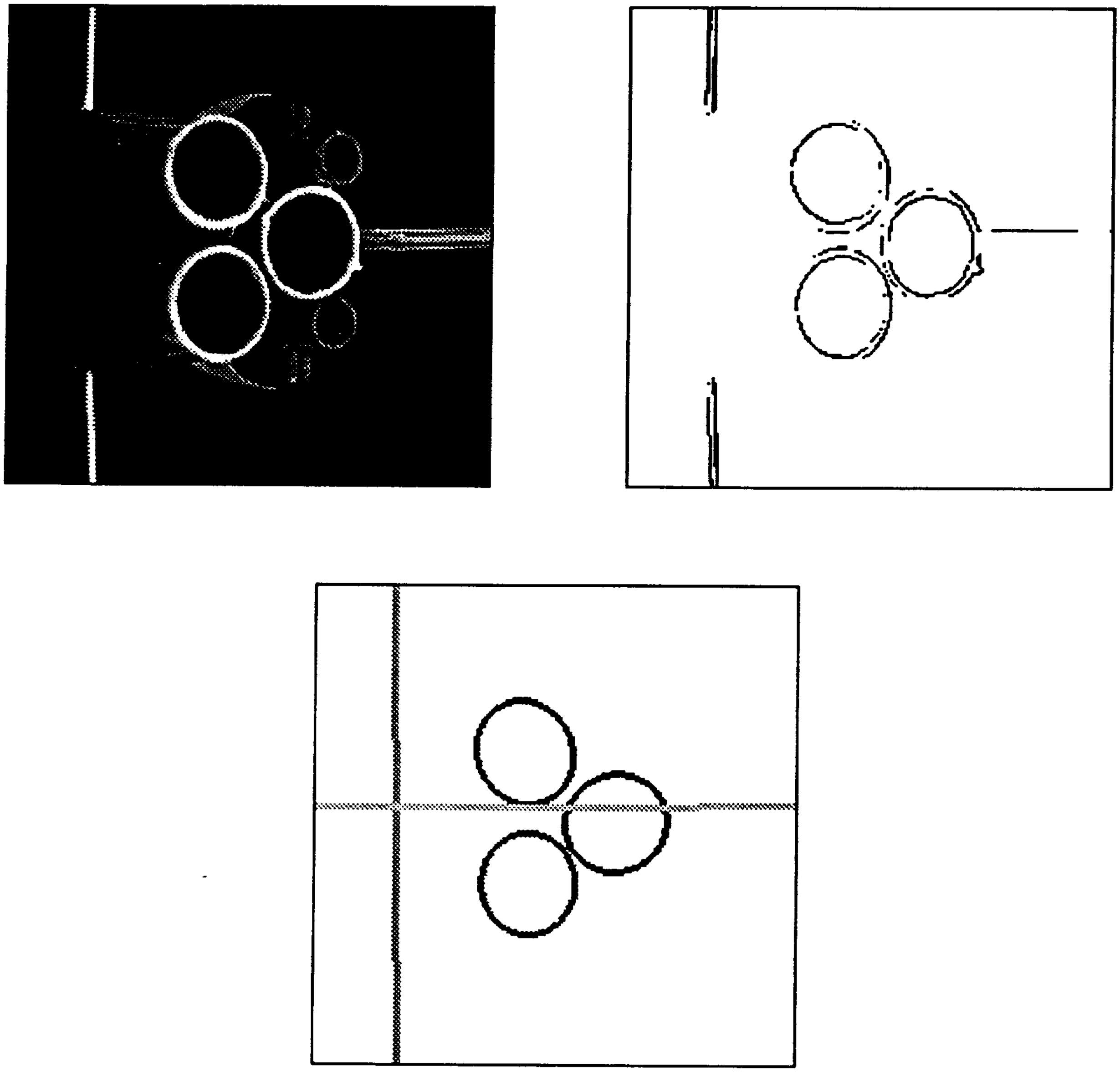


Fig. 2

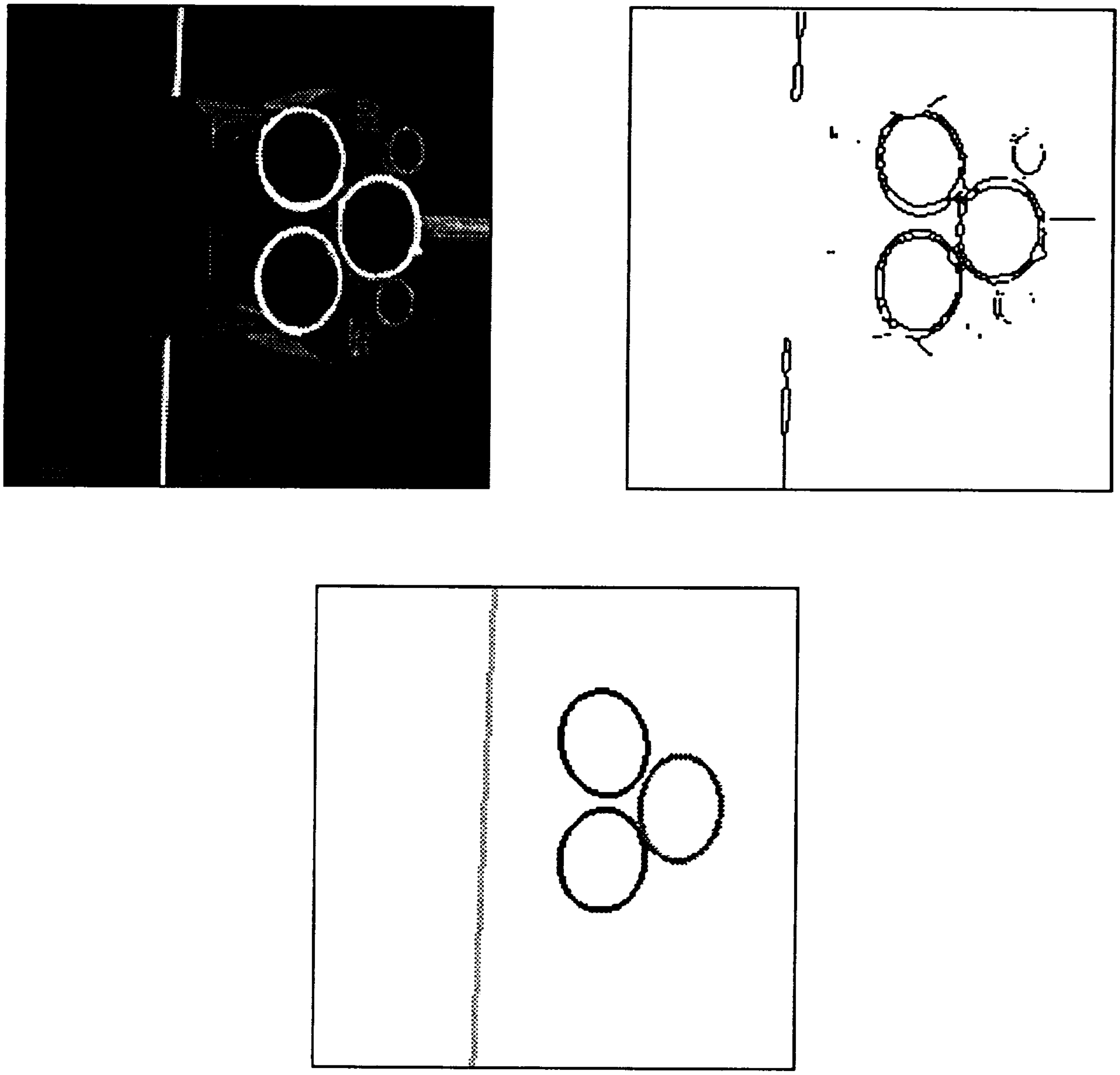


Fig 5

Acquisition of Images

We have digitized several frames from a promotional video tape on the space program. The example which was presented in the feature calculation section is one of those images. In addition, we have constructed a device to hold a model of the space shuttle at a known orientation so that we can digitize it for the "pose estimation" research. We have shown several examples of those images in an earlier report. The report on camera calibration is included. These calabration equations are necessary to accurately determine the pose parameters for the model as they appear in an image.

Recently, we have received a tape from Lincom with many simulated shuttle images in different orientations. We are in the process of devising experiments utilizing these images.

Finally, we have arranged to borrow several video tapes of shuttle missions from the NASA library (we only recently found out such a facility existed). These, we hope, will supply us with a good set of real images to test our algorithms.

Camera Calibration

The most direct way in obtaining the image coordinates (x,y) of a world point w is to apply the set of matrix equations defining the various parameters of the camera. These parameters are the focal length, offsets, and angles of pan and tilt. The mentioned parameters could be measured directly, or we could use the camera as a measuring device to estimate the parameters. This technique is know as camera calibration.

The advantage of such a technique is that it eliminates the need to keep on recalibrating the camera when the camera is moved.

Procedure:

Define a matrix A containing all the camera parameters:

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}$$

Let W be a point in the Cartesian world coordinate system

$$W = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

The homogeneous counterpart is defined as

$$W_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix}$$

Let C represent the Cartesian coordinates of any point in the camera coordinate system

$$C = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dfrac{\lambda X}{(\lambda - Z)} \\ \dfrac{\lambda Y}{(\lambda - Z)} \\ \dfrac{\lambda Z}{(\lambda - Z)} \end{bmatrix}$$

or the corresponding homogeneous image plane vector form

$$C_h = \begin{bmatrix} kx \\ ky \\ 0 \\ k \end{bmatrix}$$

Letting k=1 we can write

$$\begin{bmatrix} C_{h1} \\ C_{h2} \\ C_{h3} \\ C_{h4} \end{bmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The camera coordinates in Cartesian form are given by

$$x = \frac{C_{h1}}{C_{h4}} \; ; \quad y = \frac{C_{h2}}{C_{h4}}$$

We substitute them into the above equations. After expanding we receive

$$xC_{h4} = a_{11}X + a_{12}Y + a_{13}Z + a_{14}$$

$$yC_{h4} = a_{21}X + a_{22}Y + a_{23}Z + a_{24}$$

$$C_{h4} = a_{41}X + a_{42}Y + a_{43}Z + a_{44}$$